

# abc

N°  
49

COURS  
D'INFORMATIQUE  
PRATIQUE  
ET FAMILIALE

## INFORMATIQUE



Musiques numériques

Piraterie informatique

Olivetti M10

Softsel, grossiste mondial

EDITIONS  
**ATLAS**

**Page manquante  
(publicité et colophon)**



# Fabricants de musiques

Voici le premier d'une série d'articles où nous allons examiner le dispositif MIDI (Musical Instrument Digital Interface). Mais qu'est-ce que la manipulation sonore numérique?

La musique et les sciences exactes sont étroitement liées. Déjà, Pythagore avait pesé un ensemble de marteaux de forgeron afin de découvrir pourquoi ils semblaient être accordés lorsqu'ils heurtaient l'enclume. Il découvrit qu'un marteau dont le poids correspondait à la moitié d'un autre produisait par rapport à ce dernier un son d'une fréquence double, une octave plus haut. Cela établit le premier principe gouvernant les relations tonales en musique.

Au Moyen Age, les compositeurs mettaient au point les sons de leurs productions musicales (en particulier les motets, qui sont des chants d'église à plusieurs voix) avec une précision, dans les rythmes et les intensités, semblable à celle exigée pour bâtir les églises elles-mêmes. Ces musiques étaient parfois si compliquées que certains n'hésitaient pas à dire que seules les oreilles de Dieu pouvaient les comprendre : elles étaient magiques comme des formules mathématiques!

Plus près de nous, qui n'a pas entendu, au cours d'un concert, les musiciens commençant à compter — 1, 2, 3, 4 — à mi-voix avant de commencer à jouer?

Il est donc normal que l'univers de l'informatique soit associé à la musique. Le développement de MIDI (Musical Instrument Digital Interface) suscite actuellement beaucoup d'enthousiasme. Cette unité est conçue de façon à permettre à tout système numérique, y compris les micro-ordinateurs, de commander les fonctions d'un autre. Puisque la majorité des instruments musicaux électroniques produits actuellement sont numériques, cela laisse entrevoir d'innombrables possibilités pour les utilisateurs de micros domestiques.

Mais MIDI n'est pas une boîte magique. Il ne transformera pas en un jour un propriétaire de micro en un Berlioz ou un Stevie Wonder. Les connaissances musicales et l'imagination seront toujours indispensables, que la musique soit jouée sur un ensemble de synthétiseurs interfacés ou sur une guitare acoustique.

Afin de comprendre avec quel type d'instruments musicaux MIDI peut établir une interface, et comment apparut la musique électronique, nous devons revenir un demi-siècle en arrière. Bien avant la Seconde Guerre mondiale, des musiciens avaient commencé à expérimenter de simples générateurs sonores. Il s'agissait de dispositifs électriques qui faisaient vibrer une lamelle métallique produisant un son continu et dont la tonalité pouvait varier. Ce son fut souvent utilisé dans les bandes musicales des films de



science-fiction des années cinquante pour créer une atmosphère futuriste. Il est toujours produit par les haut-parleurs des téléviseurs pour signaler aux téléspectateurs que les émissions sont terminées.

Mais la Seconde Guerre mondiale provoqua des progrès énormes en électronique, dont l'invention de l'enregistreuse par les Allemands qui permit aux musiciens de créer et de manipuler le son d'une manière bien différente. Pour ce faire, on collait des sections de bande d'enregistrement analogique sur lesquelles un son (de la musique ou autre chose) avait préalablement été enregistré. Ces fragments de bande étaient laborieusement assemblés afin de produire un collage

## Musique obscure

La musique contemporaine demande une nouvelle notation. Les partitions de Stockhausen ou de Boulez, avec leurs représentations picturales des sons et leurs directives graphiques de minutage/synchronisation, n'ont rien à voir avec les partitions classiques, et ressemblent beaucoup plus à des diagrammes de circuits électriques. (Cl. Marcus Wilson-Smith.)



**Ron qui ?**

Le « Frère Jacques » de la musique électronique est incontestablement le thème du film *Doctor Who*, écrit pour le BBC Radiophonic Workshop en 1962 par Ron Grainer (que l'on voit ici en compagnie de ses collègues de la BBC). (Cl. BBC Television.)



d'événements sonores. Cette nouvelle musique brisait toutes les règles de la théorie musicale conventionnelle. Cette création fascinait certaines personnes et en torturait d'autres.

Au même moment, des dispositifs de variation et de distorsion ainsi que de filtrage et de modulation devinrent largement diffusés et de plus en plus contrôlables. Pendant les années cinquante, des compositeurs, comme Stockhausen en Allemagne, travaillaient dans de petits studios reliés à des stations radiophoniques locales et produisaient de la véritable musique électronique. A Paris, avec l'aide d'ingénieurs du son de l'O.R.T.F., Pierre Schaeffer introduisit ce qu'il nomma « musique concrète », un collage musical incorporant des sons quotidiens.

En Amérique, Bell Telephone Laboratories construisait ce qui était probablement le premier synthétiseur. Il occupait plusieurs pièces, et sa vocation première était d'étudier la synthèse de la voix humaine. La compagnie savait que les opératrices du téléphone, qui venaient des différentes régions des États-Unis, comprenaient très souvent mal les différents accents de leurs interlocuteurs. Le résultat se traduisait par de nom-

**Un pionnier ?**

Probablement mieux connu pour son travail avec Roxy Music au début des années soixante-dix, Brian Eno était un pionnier dans l'utilisation des premiers ordinateurs. Après avoir quitté le groupe en 1973, Eno fut un acteur important dans la musique électronique d'avant-garde. Il a aussi collaboré avec des célébrités comme David Bowie. Plus récemment, Eno a travaillé sur des partitions de musique de télévision et de cinéma.



breuses erreurs de transmission, de faux numéros... Bell espérait donc, de façon un peu naïve pour l'époque, qu'une seule et même voix — synthétisée — puisse résoudre tous ces problèmes. De nombreux musiciens américains reçurent leur formation électronique à ce moment, grâce à la compagnie Bell. En Grande-Bretagne, un travail similaire était en cours, mais à une échelle moins ambitieuse.

La première tentative de musique sur ordinateur fut menée en 1957, lorsque Lejaren Hiller entra une série d'instructions dans l'ordinateur de l'université de l'Illinois. Ces instructions se répartissaient en quatre groupes de données techniques, qui furent alors traduites en notation musicale. Le résultat fut un morceau en quatre mouvements pour quartet à cordes intitulé « La suite Illiac ». La musique elle-même, bien que correctement arrangée pour un violoncelle, une viole et deux violons, n'avait que très peu de caractère.

Quelques années plus tard, Hiller créa une autre œuvre, utilisant cette fois l'ordinateur IBM 7090. Il conçut un schéma de programmation nommé MUSICOMP (MUSIC SIMULATOR for COMPOSITION PROCEDURES), qui autorisait une plus grande souplesse dans le processus de composition finale. Il le nomma « Computer Cantata » ; ce morceau était écrit pour un chanteur, accompagné de sons électroniques enregistrés. De nouveau, la musique n'offrait qu'un intérêt passager. Mais Hiller avait démontré à ses collègues musiciens qu'un ordinateur pouvait effectivement être utilisé de façon créative.

Ses travaux ne représentent qu'une partie de la quantité colossale de recherches qui furent menées dans les universités américaines au cours des années qui suivirent. John Chowning, un autre pionnier, utilisa plus tard un ordinateur pour comprendre comment la perception du son était modifiée en déplaçant sa source d'un endroit à un autre. L'utilisation par Yamaha de ses travaux de recherche contribua à la création du type de synthétiseur produit au milieu des années quatre-vingt.

A l'exception de la musique de science-fiction, la musique électronique demeura dans le royaume de la musique classique pendant plusieurs années. Le public devint ainsi de plus en plus conscient du changement apporté par l'approche et la technique retenues par les compositeurs d'avant-garde. Un concert de musique contemporaine typique des années soixante nécessitait plusieurs exécutants, certains jouant des instruments conventionnels, d'autres étant impliqués dans le traitement du son produit par ces instruments. Tous les exécutants, y compris les « techniciens », devaient suivre une partition, mais sa notation était fort éloignée de celle d'une partition traditionnelle.

En plus de directives nouvelles comme celles qui décrivent les positions des microphones et les variations de filtrage, les compositeurs avaient tendance à donner des directives visuelles pour décrire l'apparence de ces nouveaux sons. Dans certains cas, la partition ressemblait à une feuille



de travail de graphiste. Ce problème — comment diriger l'exécution musicale, quel langage utiliser, et comment visualiser avec précision le résultat — existe toujours dans les systèmes musicaux numériques des années quatre-vingt.

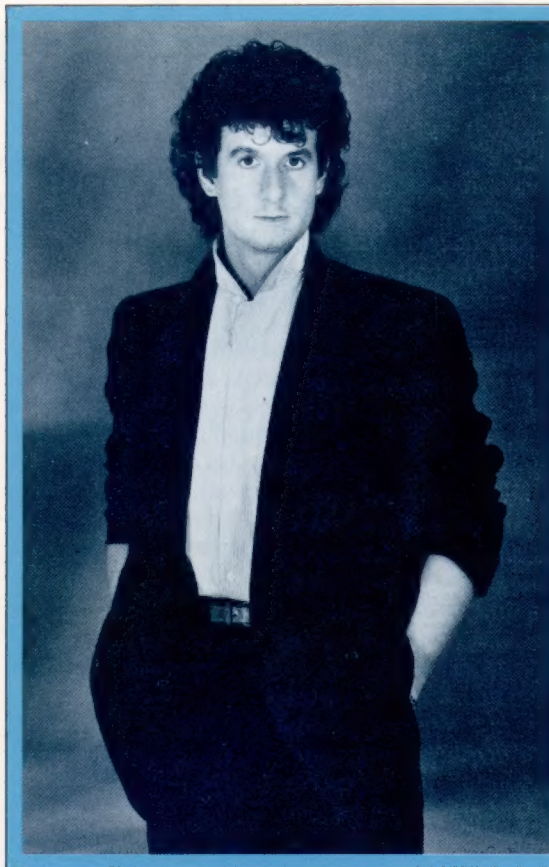
Au cours des années soixante, les musiciens pop commencèrent à passer de plus en plus de temps dans les studios d'enregistrement, et commencèrent aussi à expérimenter la musique électronique. L'exemple classique est celui des Beatles, qui trouvèrent en George Martin non seulement un excellent ingénieur d'enregistrement, mais aussi un musicien qui était au fait des récents développements dans le domaine classique. Il encouragea les Beatles à utiliser le studio entier comme instrument de musique, et le groupe commença peu de temps après à utiliser des techniques de collage de bande et à incorporer le son synthétisé.

Certains musiciens firent leur renommée en utilisant des unités de traitement sonores particulières. Un guitariste comme Jimmy Page modela le son en se basant sur celui des musiciens noirs américains des années quarante, puis en utilisant une série de commandes de distorsion; il produisit le son caractéristique de Led Zeppelin. Avec l'utilisation que fit Jimi Hendrix du braillement « feedback » et de la pédale « wah-wah », tous ces événements annonçaient le début du hard rock.

Dans les années soixante-dix, les diverses unités de traitement et de génération sonores qui étaient offertes depuis 1950 incorporèrent le transistor. Elles devinrent plus petites, plus portables, et ne furent donc plus limitées à rester dans un studio. Les guitaristes pouvaient jouer en concert en utilisant une variété d'effets spéciaux. Un peu plus tard, les organistes et les pianistes purent s'offrir des synthétiseurs qu'ils pouvaient utiliser sur scène.

Ces synthétiseurs comportaient un ensemble d'oscillateurs en syntonie, de définisseurs d'enveloppe (pour créer les caractéristiques d'attaque, de maintien et de chute du son), des filtres variables, des modulateurs d'anneaux (qui pouvaient donner de nouvelles fréquences aux signaux), et des générateurs de bruit. Tout comme Jimi Hendrix a été un modèle pour les guitaristes, Brian Eno devint celui des joueurs de synthétiseur, principalement en raison de ses liens avec la « nouvelle musique » de l'avant-garde classique.

En même temps, l'équipement de studio d'enregistrement devint plus perfectionné, puisque les musiciens comptaient sur le studio pour obtenir ce qu'ils ne pouvaient produire sur scène. Le pupitre de mixage, permettant aujourd'hui de placer des enregistrements successifs sur 16 ou 24 pistes de bande, était toujours trop grand pour être transporté, et la plupart des unités de traitement étaient longues à installer. Aux États-Unis et en Grande-Bretagne, une nouvelle génération de producteurs fit son apparition. Ils avaient souvent démarré en tant qu'ingénieurs et connaissaient l'équipement plus à fond que les musiciens qui les payaient pour obtenir le « bon son ».



**Un grand producteur**  
Producteur du groupe britannique Culture Club, Steve Levine doit surtout sa célébrité à son talent pour associer la musique électronique aux voix humaines. Levine fut le premier producteur britannique à utiliser la batterie Linn, un synthétiseur de percussions programmable, et il a récemment élevé les techniques d'enregistrement numérique au niveau artistique. Levine a beaucoup utilisé des synthétiseurs et d'autres équipements musicaux numériques dans sa dernière création, *Believin*, écrite en collaboration avec Boy George.

En Jamaïque, des ingénieurs commencèrent à employer le pupitre de mixage comme un instrument. Les chansons terminées, enregistrées sur des bandes à pistes multiples, pouvaient être reconstituées en pistes rythmiques individuelles. Les diverses contributions vocales et instrumentales pouvaient être réutilisées, en tant que matériel brut, pour d'autres arrangements.

L'arrivée de synthétiseurs numériques introduisit la possibilité de coder des sons non électroniques. On nomme ce processus « échantillonnage ». Les instruments de percussion comme le Linn devinrent des instruments de studio recherchés et firent rapidement partie du matériel de scène. Au milieu des années quatre-vingt, l'échantillonnage et la manipulation du son deviennent un véritable art, et les équipements de studio et de scène comprennent généralement plus d'articles numériques que d'instruments analogiques. Des groupes très populaires, comme Culture Club, combinent leurs propres connaissances de composition aux techniques de producteurs comme Steve Levine, qui utilisent des instruments et des unités de traitement qui valent des centaines de milliers de francs.

La nécessité d'une interface pouvant relier un instrument à un autre ou accroître le potentiel d'un synthétiseur en incorporant le système d'exploitation et la mémoire d'un micro-ordinateur, suscita un regroupement des fabricants d'instruments de musique. Ils créèrent ainsi la norme MIDI en avril 1983, et, depuis, peu de sociétés ont osé annoncer un nouveau synthétiseur qui ne soit pas compatible avec cette norme.

# Lignes d'erreur

Une logique mal articulée peut avoir des résultats graves. En particulier, aux points d'interface entre les sous-programmes et entre le programme et son utilisateur.

Il existe de nombreuses sources potentielles d'erreurs à chaque étape de la création d'un programme, depuis sa conception et sa codification jusqu'aux tests. Il faut reconnaître que les erreurs sont principalement introduites au moment de la définition et de la conception, et ce, dans la mesure où la réflexion sur la nature des problèmes à résoudre et les précautions à prendre auront été insuffisantes pour s'assurer que le programme va correctement répondre à ce que l'on attend de lui.

Nous pouvons réduire les risques d'erreur en suivant les méthodes de conception structurées que nous avons déjà rencontrées. Il est possible que d'autres erreurs se glissent lorsque le travail de conception est traduit en code — des erreurs de frappe peuvent introduire des problèmes, comme le savent tous ceux qui ont déjà fait une faute dans une variable. Même les étapes de vérifi-

cation et de mise au point peuvent causer d'autres erreurs, puisque la correction d'une erreur peut entraîner l'apparition d'une autre.

Mais c'est aux points d'interface — entre les sous-programmes et entre le programme et son utilisateur — que se situent la plupart des erreurs. Une attention particulière doit être exercée pour s'assurer que toutes les valeurs transmises par ces interfaces sont correctes et qu'elles se situent à l'intérieur de la plage requise par le programme. Les valeurs peuvent être vérifiées par la routine qui les transmet, ou par celle qui les accepte.

Ainsi, des vérifications doivent être menées lorsque la sortie dépend d'une valeur entrée par un utilisateur ou lue dans un fichier. Les valeurs qui sont entrées dans une routine doivent toujours être vérifiées. Les sous-programmes peuvent être conçus de façon à donner un ensemble de sorties clairement définies; mais les êtres humains ne fonctionnent pas si méthodiquement, et ils ont tendance à donner de nombreuses réponses différentes à un message de sollicitation. Des vérifications sévères doivent donc être faites dans toutes les routines qui acceptent des données en provenance des utilisateurs. Similairement, les fichiers de données peuvent être détériorés ou mal lus, des vérifications doivent donc être effectuées dans chaque routine de manipulation de fichiers.

Les erreurs entraînent rarement l'interruption des programmes. Lorsque c'est le cas, c'est que le programme a enfreint une règle du langage (utilisation illégale d'un opérateur, par exemple, comme dans `RESULTAT = PREM$ + SECOND$`), ou une règle du système d'exploitation (ouverture d'un trop grand nombre de fichiers en même temps, par exemple). Le code suivant peut sembler constituer un programme parfaitement valide :

```
10 FOR COMPTE = 1 TO 10
20   SOM = SOM + 1
30   PRINT COMPTE, SOM
40   GOTO 10
50 NEXT COMPTE
```

Il s'agit cependant d'un algorithme qui ne se termine pas et qui bloquera le système en raison du mode de fonctionnement du langage. Dans ce cas, le langage (BASIC) utilise la pile pour gérer les boucles `FOR...NEXT`, incrémentant la pile au début de chaque nouvelle boucle. Dans ce programme, la ligne 50 (avec la commande `NEXT` qui décrémenterait la pile) n'est pas atteinte, et la pile se remplit graduellement jusqu'à ce qu'un mes-

## Liste de contrôle d'erreur

Une approche logique structurée est essentielle pour éviter les erreurs; la liste de contrôle d'erreur suivante (par G.J. Myers dans « The Art Of Software Testing ») illustre une telle approche.

### Les variables

1. Les noms de variables sont-ils tous uniques? En se souvenant que de nombreux interpréteurs n'utilisent que les deux premiers caractères.
2. Une variable (spécialement les compteurs de boucle ou les paramètres de sous-programme) a-t-elle été réutilisée pendant que son contenu reste significatif?
3. Les indices des tableaux sont-ils légaux et sont-ils des nombres entiers?
4. Les indices des tableaux commencent-ils à l'élément zéro ou à l'élément un?

### Calculs

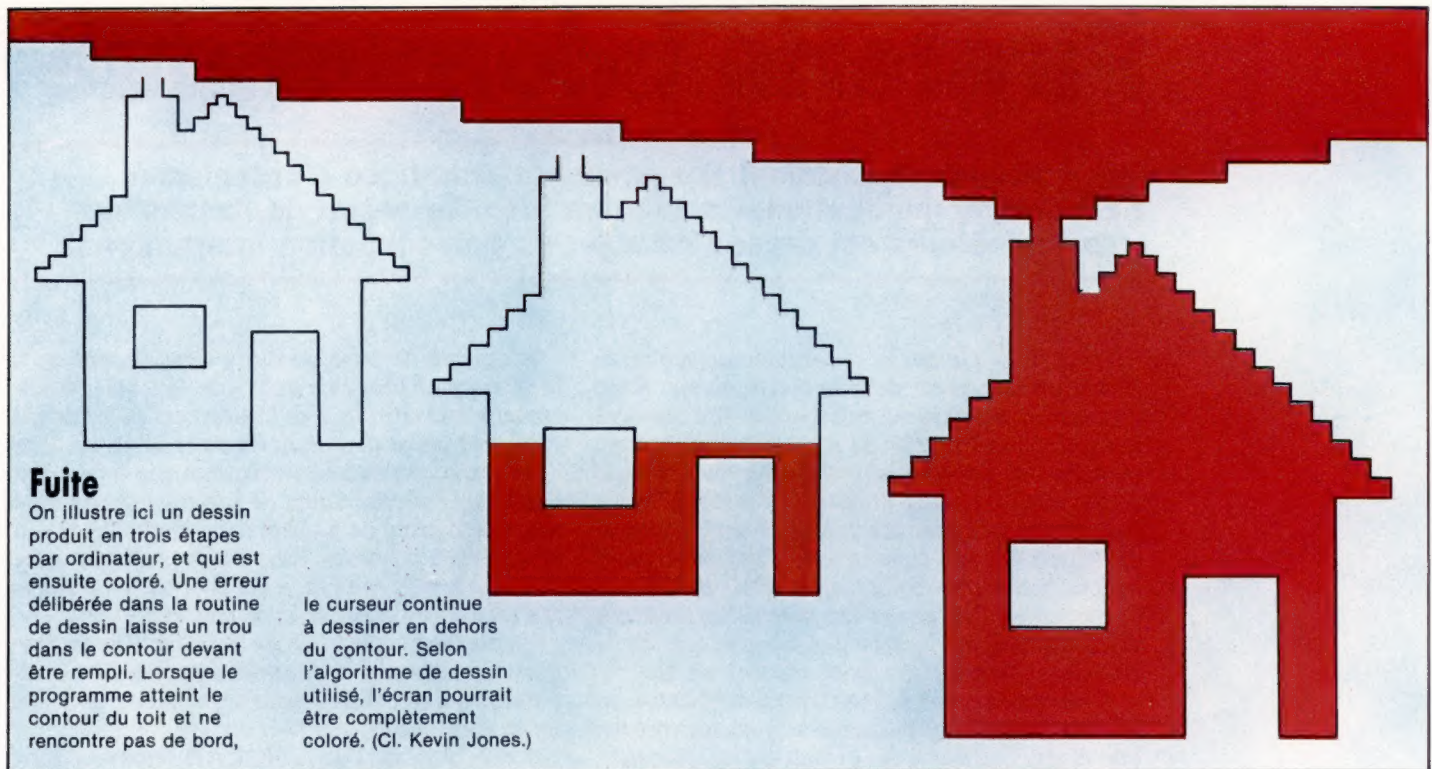
1. Les calculs donnent-ils des résultats de chaîne ou des résultats numériques, et ces résultats sont-ils affectés à des variables de chaîne ou à des variables numériques?
2. Un calcul donne-t-il un résultat trop petit ou trop grand pour permettre sa manipulation par l'ordinateur? Cela peut-il causer une erreur de division par zéro?
3. Les erreurs d'arrondi peuvent-elles être significatives?
4. Toutes les opérations d'une expression sont-elles exécutées dans l'ordre logique?

### Comparaisons

1. Les chaînes sont-elles comparées avec des chaînes, les nombres avec des nombres?
2. L'emploi de majuscules ou de minuscules est-il significatif?
3. Des chaînes de longueurs inégales sont-elles comparées, et les différences de longueur ont-elles plus ou moins d'importance que les différences de caractères?
4. Les opérateurs de comparaison et les opérateurs booléens sont-ils mixés adéquatement?
5. La préséance des opérateurs booléens et de comparaisons affecte-t-elle l'exécution des expressions de comparaison?

### Contrôle

1. Les boucles et les algorithmes prennent-ils fin, quel que soit l'état des variables?
2. Chacune des boucles et des routines n'a-t-elle qu'un seul point d'entrée et de sortie?
3. Lorsqu'une instruction `IF...THEN` est négative, le déroulement du programme se poursuit-il à la prochaine instruction du programme ou à la ligne suivante du programme?
4. Qu'arrive-t-il si aucune des conditions de test d'une instruction à branchements multiples n'est satisfaite?



sage de dépassement en capacité de pile soit produit, et que l'interpréteur interrompe le programme. De telles erreurs sont généralement facilement repérées.

Un type d'erreur plus insidieux permet au programme d'être exécuté normalement, mais invalide les résultats. A titre d'exemple, nous avons choisi d'examiner un motif qui dessine une forme sur l'écran et qui la colore. Lorsqu'un bord est atteint, l'ordinateur fait faire demi-tour au curseur et continue jusqu'à ce qu'il atteigne un autre bord. Pour permettre le fonctionnement d'une telle routine, les bords doivent être complets et bien définis.

Les versions du langage BASIC utilisées par la plupart des micros domestiques facilitent la correction d'erreurs en produisant des messages clairs et concis, et en permettant de relancer un programme après avoir modifié les valeurs des variables au clavier — une fonction très pratique lors de la mise au point d'un programme. La plupart des dialectes BASIC permettent l'utilisation d'une commande semblable à `ON ERROR GOTO` pour passer la main à une routine spéciale de gestion d'erreur, ce qui permet à l'utilisateur de se consacrer à des erreurs plus graves. Pour ce faire, il suffit d'inclure une ligne de programme comme celle-ci :

```
30 ON ERROR GOTO 20000: REM routines de gestion d'erreur
```

au début du programme. Toute erreur entraînera donc l'exécution de l'instruction `GOTO 20000`. `ON ERROR` définira aussi généralement deux variables; la première stocke un numéro d'erreur qui indique quel type d'erreur est survenu, et l'autre renferme simplement le numéro de la ligne où l'erreur a été rencontrée. Les noms donnés à ces

variables et les numéros d'erreur qui en résultent varient d'une machine à l'autre; le manuel doit donc être consulté. Dès qu'une erreur survient, le déroulement du programme est détourné vers la ligne 20000; l'erreur est identifiée à partir du numéro contenu dans la variable associée, et l'intervention adéquate est entreprise.

Un programme bien écrit ne comportera pas plus d'une routine `ON ERROR`. Une telle routine ne pourra gérer les erreurs de syntaxe, les insuffisances de mémoire, les dépassements de capacité de pile, etc. Le principal avantage qu'offre cette fonction consiste à mettre à l'arrêt le système de façon conforme, en s'assurant que tous les fichiers sont fermés, et que l'utilisateur sait exactement ce qui est arrivé.

- La commande `ON ERROR GOTO` et le passage subséquent au programme principal constituent une entrée et une sortie supplémentaires pour une routine. Cela va à l'encontre d'un des principes fondamentaux de programmation qui veut que les routines ne comportent qu'un point d'entrée et qu'un point de sortie.

- La protection contre une division par zéro doit être placée dans la routine qui effectue la division. Il n'est pas recommandé de concevoir des algorithmes qui peuvent bloquer le système. Si la vérification d'erreur supplémentaire ralentit le programme à un niveau inacceptable, la routine doit être conçue à nouveau afin d'éliminer ce problème.

- Les routines de gestion d'erreur deviennent rapidement des chaînes compliquées d'instructions `IF...THEN...ELSE` créant des sorties multiples. Elles dépendent inévitablement de la numérotation des lignes du reste du programme et doivent donc être réécrites lorsque l'une des routines qui les utilise est à nouveau conçue.



# Piraterie informatique

Des « pirates » qui violent le système informatique d'organismes d'État ou de multinationales existent bien. Le secret de l'accès aux gros ordinateurs est devenu un sujet de préoccupation international.

Le film *War Games* a certainement passionné plus d'un utilisateur de micro-ordinateur. Avec un micro et un modem, le héros du film parvient à appeler l'ordinateur de son université et à en modifier les données pour falsifier ses résultats d'examens. Le pirate avait en outre forcé l'accès à d'autres systèmes informatiques pour réserver des billets d'avion et recopier un logiciel exclusif de jeux. Les choses commencèrent pourtant à mal tourner pour lui, lorsqu'il commit l'imprudence de s'attaquer au système informatique de la NORAD (North American Air Defence).

De nombreux cas de piraterie informatique ont effectivement lieu; les fraudeurs sont souvent des adolescents utilisant des ordinateurs personnels.

L'origine de cette piraterie tient davantage à la négligence humaine qu'à une véritable défaillance informatique. Les utilisateurs patentés du réseau Arpanet disposent de mots de passe. Il se trouve que ces derniers n'étaient pas très originaux. Nos petits malins de pirates ont suspecté que l'université de Californie à Berkeley devait être reliée à Arpanet. Ils ont alors essayé le mot de passe le plus évident, « UCB », et ils ont réussi à s'introduire ainsi dans le réseau tout entier.

Cependant, si le Q.G. de la NORAD fait bien partie du réseau de données Arpanet, les ordinateurs qui ont en charge la surveillance aérienne ne le sont pas.

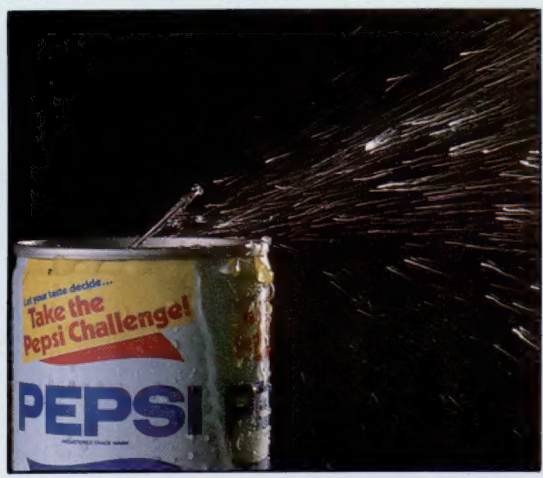
Les ordinateurs de la NORAD semblent donc à l'abri des intrusions informatiques, mais cela n'est pas le cas de tous les systèmes de traitement importants. Ainsi, en 1983, un groupe d'adolescents de Milwaukee a réussi à accéder à divers ordinateurs appartenant à des universités, des sociétés et au Los Alamos National Laboratory, qui participe à la conception et à la fabrication d'armements. Mais là encore, selon les autorités, aucune information confidentielle n'a été découverte.

Ce ne sont là que quelques-uns des cas connus de piraterie informatique. Mais de nombreux cas n'ont probablement jamais été portés à la connaissance du public; peu de détenteurs de gros systèmes informatiques, censés être confidentiels, sont désireux de reconnaître la vulnérabilité de leur outil si cher payé... Mais comment savoir si un utilisateur non habilité, ou un imposteur, s'est « branché » sur un système à un moment donné? En fait, ce sont souvent les « pirates » eux-mêmes qui signalent leur passage par des messages du genre « Devinez qui je suis? », ou en signant par exemple : « Celui-qui-fait-se-planter-le-système ».

Comment se fait la connexion « sauvage »? Le futur pirate a simplement besoin d'un micro-ordinateur, d'un modem et d'un peu d'ingéniosité. Le premier obstacle à franchir est de trouver le numéro de téléphone de l'ordinateur à « déployer ». Pour les réseaux publics de commutation de données, comme Transpac ou Télétel, c'est chose facile, puisqu'il existe des annuaires publics. Pour les ordinateurs privés, c'est plus difficile. Vous pouvez essayer, en vous basant sur la localisation de l'ordinateur visé, de trouver son code téléphonique. C'est la technique utilisée par le pirate informatique du film *War Games*. Il avait programmé son micro pour appeler tous les numéros de téléphone possibles de sa ville. Lorsqu'un ordinateur répondait

## Distribution illégale

Même si la société américaine Pepsi-Cola nie toujours l'incident, il semble bien qu'elle ait été victime d'une piraterie informatique. Un fraudeur se serait mis en contact avec un ordinateur de la société au Canada. La fraude aurait consisté à faire expédier de grosses livraisons de Pepsi-Cola à des endroits précis afin de réussir des transferts de fonds.



Ian McKinnell

Les systèmes informatiques « victimes » vont depuis les gros ordinateurs appartenant à des universités ou à de grosses sociétés, jusqu'à des associations d'utilisateurs de micros et prestataires de services. Tout ordinateur accessible par téléphone est vulnérable. Ainsi, la réalité a rejoint la fiction en 1983, quand la force aérienne américaine (NORAD) a soupçonné qu'une piraterie avait eu lieu à l'encontre de son système informatique de Omaha, Nebraska.

Les responsables, deux adolescents de Los Angeles, avaient réussi à s'introduire dans le réseau informatique secret du département de la Défense américain, Arpanet. Utilisant un Commodore Vic-20 et un Tandy TRS 80, ils étaient parvenus à interroger plusieurs ordinateurs reliés à Arpanet, appartenant à des fournisseurs, à des organismes de recherche et à des universités en rapport avec l'armée.





— identifié par un sifflement bien particulier semblable à un klaxon — le micro-ordinateur relevait le numéro. Par contre, lorsqu'un particulier répondait, le modem raccrochait et passait au numéro suivant. Avec un système de recherche automatique, cela est très rapide. Mais lorsqu'il faut faire soi-même les numéros, le jeu devient assez fastidieux...

Une fois connecté à l'ordinateur recherché, un mot de passe est demandé. Certains systèmes autorisent cependant un accès restreint si l'on répond « INVITÉ » ou « NOUVEAU-venu », ou encore si l'on tape simplement un retour-chariot. Mais le véritable pirate s'ingéniera certainement à trouver le mot de passe. Souvent, ce n'est pas très difficile, car les utilisateurs manquent d'imagination en utilisant des noms comme « DUPONT », des mots assez évidents comme « SECRET » ou même « MOT DE PASSE ». Les mots de passe constitués de chiffres sont également assez faciles à découvrir car l'utilisateur choisit des séquences de chiffres facilement mémorisables comme, par exemple, une date de naissance : « 090560 ». Les ordinateurs autorisent plusieurs essais pour le mot de passe. Dans ce cas, vous pourrez toujours rappeler pour continuer vos essais, sans que l'ordinateur-hôte ne vous suspecte.

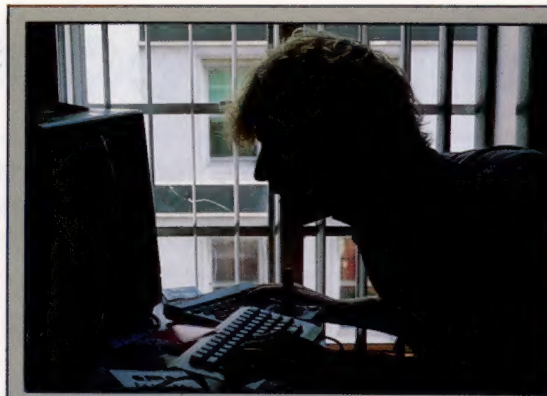
Une fois admis dans le système, le pirate « ordinaire » se contentera le plus souvent de consulter un peu tous les fichiers. Certains pirates rageurs détruisent des fichiers, laissent des messages obscènes ou s'efforcent de rendre inopérant le système tout entier !

Les ordinateurs les plus sophistiqués comportent souvent des accès directs en cas d'urgence. Les informaticiens peuvent ainsi entrer dans un programme sans avoir à passer par les protections. Mais les utilisateurs de programmes amateurs ne savent généralement pas qu'une telle porte dérobée existe.

Vous avez sans doute remarqué que les cas mentionnés ici impliquent d'une manière ou d'une autre des ordinateurs d'universités. Cela est dû à ce que ces derniers, outre le fait qu'ils sont accessibles par téléphone, sont basés sur le principe d'un accès largement ouvert. L'inconvénient est évidemment la facilité de piratage. Cette porte ouverte a en outre l'inconvénient supplémentaire de permettre l'accès à d'autres systèmes qui sont en liaison avec les gros ordinateurs, en se faisant passer pour un utilisateur légitime.

Quelle est la motivation de ces pirates d'un nouveau genre ? C'est, la plupart du temps, le seul défi de battre le système. De nombreux fraudeurs se donnent une sorte de code moral implicite qui est de ne pas causer de dommages aux fichiers, ni de faire de la provocation gratuite et violente. Il s'agit pour eux d'un jeu, dont le piquant est tout entier dans le déchiffrement des codes secrets des systèmes centraux.

Les banques ont longtemps subi les exactions informatiques, mais il y a encore peu de temps, il s'agissait surtout de détournements internes dus à leur propre personnel. Des employés malhonnêtes organisaient des transferts de fonds sur des



La démarche utilisée pour pirater un ordinateur relève essentiellement de la méthode d'essais et d'erreurs. Cela aboutit très rarement. En effet, même si un fraudeur parvient à localiser le système, il se heurte alors à la question du mot de passe qui lui est posée avec insistance.

Parfois, à dessein ou par hasard, un utilisateur trouve un mot de passe valide de priorité maximale. Il peut alors accéder au système correspondant. Le dialogue suivant montre comment on accède à des informations relatives aux utilisateurs légaux de l'ordinateur.

```

CONNEXION 9X001001
14.12
12/77/84
>
UTILISATEUR?
>ASSISTANCE
UTILISATEUR?
>#OFF
UTILISATEUR?
>UK001
MOT DE PASSE?
>INVITE
MOT DE PASSE?
>NOUVEL UTILISATEUR
MOT DE PASSE?
>AZERTY
LOGOFF 15.13 TEMPS
DE CONNEXION = 0.13 MN

BYF990
15.14.02 12/07/84
>
UTILISATEUR?
>UK001
MOT DE PASSE?
>OPSYS
LOGON 15.15.07 12/07/84
NOTE:
BYF990/
UTILISATEUR:
UK001
N° SERIE : ZAL90-7
N° série du logiciel
PRIDRITE: SUPER UTILISATEUR
STATUS: ACTIF
VOUS ETES OPSYS
7 UTILISATEUR(S)
APP01 APP02 BYF7 BTY04
BZX00 BZX02 OPSYS
>SUPPRIMER APP01
UTILISATEUR(S) APP01
DECONNEXTE
>QUI EST BTY04
    
```

## Essais et erreurs

Tapez RC pour donner l'identité utilisateur.  
Il n'existe pas d'assistance, l'accès n'est pas libre.  
Premier essai d'identité.  
L'identité donnée ne permet pas l'accès.  
Deuxième essai d'identité.  
Identité acceptée, demande du mot de passe.  
Premier essai.  
Refusé, demande réitérée.  
Deuxième essai.  
Refusé, troisième demande.  
Essai de la dernière chance.  
L'utilisateur est déconnecté après le troisième essai refusé pour trouver le mot de passe.

Tapez RC.  
Identité acceptée, demande du mot de passe.  
Premier essai : opérateur système.

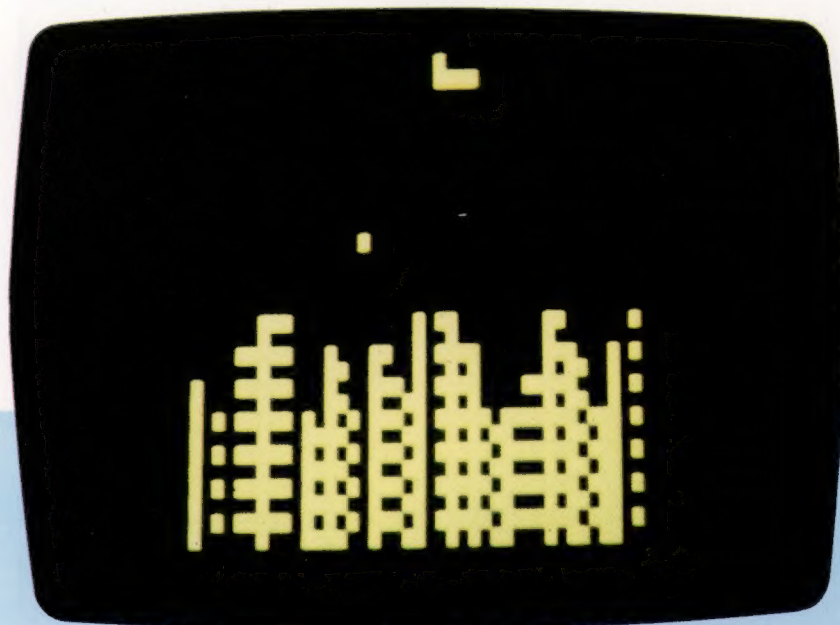
N° série du logiciel.  
Plein accès à tous les fichiers.  
Un utilisateur peut voir son accès inactif (invalidé), s'il n'utilise pas régulièrement le système.  
Confirmation.  
Liste de tous les utilisateurs.  
Commande réservée à l'opérateur du système.  
Utilisateur valide déconnecté.

comptes « bidons » par exemple. Les estimations relatives aux détournements de fonds par ordinateur varient de 300 millions à 25 milliards de francs par an pour un seul pays comme la France. Naturellement, les banques et les sociétés victimes de ce genre d'exactions ne le font pas savoir publiquement ; il est donc particulièrement difficile d'en estimer le montant exact.

Avec le développement de la micro-informatique personnelle et celui des réseaux de données utilisant le téléphone, la fraude par ordinateur ne peut que s'accroître. Qu'il s'agisse de malveillance de la part de jeunes « pirates » soucieux de nuire, supprimant des fichiers et accaparant du temps de traitement, ou de truands professionnels détournant systématiquement des sommes d'argent sur leurs propres comptes, les méthodes de base de la piraterie informatique sont les mêmes.

# Blitz avec Alice

La plupart des micros familiaux permettent de jouer avec Blitz ou quelque chose qui lui ressemble sous une autre appellation. Pierre Monsaut a écrit ce programme pour le micro Alice.



Le but du jeu est de détruire la ville que vous survolez afin de pouvoir atterrir. A chaque passage, votre avion vole un peu plus bas. Vous ne pouvez larguer une bombe (en appuyant sur une touche quelconque) que lorsque la bombe précédente a atteint son objectif ou le sol. Lorsque votre avion a atterri (ou lorsqu'il s'est écrasé contre un immeuble), le score est affiché, ainsi que le record du jour. Si ce jeu vous paraît trop difficile, vous pouvez changer les limites de la ville (6 et 26, ligne 60) et la hauteur maximale des immeubles (en remplaçant 8, ligne 80, par une valeur supérieure, par exemple 10).

```

5 REM *****
10 REM * BLITZ *
15 REM *****
16 REM
17 REM INITIALISATION
18 REM
19 REM A#=AVION
20 A#=CHR$(128)+CHR$(155)+CHR$(1
47)
25 REM B#=BOMBE
30 B#=CHR$(145)
35 REM H=POSITION AVION ET SCORE
40 H=0
50 CLS 0
55 REM AFFICHAGE VILLE
60 FOR I=6 TO 26
70 C=RND(7)+151
75 REM REMPLACER 8 PAR AUTRE
76 REM VALEUR POUR CHANGER
77 REM HAUTEUR IMMEUBLES
78 REM LIGNE 80
80 FOR J=15 TO RND(4)+8 STEP-1
90 PRINT@ J*32+I,CHR$(C);

```

```

100 NEXT J
110 NEXT I
114 REM
115 REM BOUCLE PRINCIPALE
116 REM
120 PRINT@ H,A#;
125 REM AVION ECRASE?
130 IF PEEK(16387+H)<>128 THEN 1
000
155 REM TIR
160 IF INKEY#<>" " AND B=0 THEN B
=H+33
165 REM BOMBE ATTEINT LE SOL?
170 IF B<>0 THEN GOSUB 5000
175 REM PAS DE BOMBE LACHEE?
180 IF B=0 THEN GOSUB 6000
185 REM AVANCE AVION
190 H=H+1
195 REM AVION POSE?
200 IF H=507 THEN 1000
210 GOTO 120
994 REM
995 REM AVION POSE
996 REM

```

```

999 REM RECORD BATTU?
1000 IF H>R THEN R=H
1010 PRINT@ 3,"SCORE :";H;
1020 PRINT "RECORD :";R;
1030 FOR I=1 TO 100
1040 NEXT I
1050 R#=INKEY#
1060 PRINT@ 73,"UNE AUTRE ?";
1070 R#=INKEY#
1080 IF R#="" THEN 1070
1090 IF R#<>"N" THEN 20
1100 CLS
1110 END
4994 REM
4995 REM BOMBE LACHEE
4999 REM SOL ATTEINT
5000 IF B>=510 THEN B=0:GOTO 502
0
5005 REM IMMEUBLE ATTEINT?
5010 IF PEEK(16384+B)<>128 THEN
PRINT@ B,CHR$(128);
5015 REM AFFICHAGE BOMBE
5020 PRINT@ B1,CHR$(128);
5030 IF B<>0 THEN PRINT@ B,B#;:B
1=B:B=B+32
5040 RETURN
5994 REM
5995 REM DELAI
5996 REM
6000 FOR I=1 TO 20
6010 NEXT I
6020 RETURN

```



# Bellissimo!

**Olivetti exerce depuis longtemps et avec succès des activités dans le domaine du bureau. La vocation micro-électronique de l'entreprise est aujourd'hui manifeste. Le micro portable M10 en est une preuve.**

On a longtemps évoqué les ordinateurs portables. On peut désormais parler d'ordinateurs portatifs. En effet, les premiers dépassent largement 5 kg (ils en affichent le plus souvent une bonne dizaine sur la balance); les seconds sont nettement plus discrets. Ils se glissent dans un attaché-case et ajoutent moins de kilos au bout de votre bras qu'il n'y a de doigts à une main.

Dans ce domaine, le M10 d'Olivetti fait figure de poids plume — 1,700 kg tout nu — prêt à satisfaire vos moindres désirs informatiques en toute circonstance et en tout lieu, ou presque... Certes, il faut dire immédiatement que cet équipement est essentiellement destiné aux travaux d'affaires, même si Olivetti vous propose des programmes de jeux. Il ne faut pas se tromper. Le prix de vente de base, qui tourne autour de 8 000 F, est là pour rappeler qu'il existe des matériels de loisirs plus abordables...

Le M10 est arrivé sur le marché avec d'autres ordinateurs, comme le TRS 100 de Tandy ou le PC 8201 de NEC, en apportant une véritable transformation à cette informatique qui voyage ou qui veut de plus en plus voyager. Mais peut-être avec un plus sur ses concurrents, grâce à son design où l'on reconnaît sans difficulté la patte des concepteurs italiens.

Regardons ses caractéristiques techniques. Le M10 fonctionne sur batteries, avec une autonomie de vingt heures annoncée par le constructeur, ou sur le réseau. Il dispose d'un écran de visualisation à cristaux liquides orientables. (Il est inclinable jusqu'à 30°.) Pour éviter toutes sortes de difficultés de lecture, cet écran est muni d'un variateur de contraste. La capacité d'affichage de l'écran est de 8 lignes de 40 caractères (soit 320 positions). La résolution graphique est de 64 x 240 points. Il est bien évident que, depuis sa sortie (en France, au début de 1984), d'autres matériels font déjà mieux au niveau de la capacité d'affichage. Reste, néanmoins, l'avantage de pouvoir orienter l'écran; ce que ne proposent pas tous les concurrents d'Olivetti. On peut toutefois se demander si cette option résistera longtemps aux multiples chocs qu'est appelé à subir un micro-ordinateur éminemment portable comme celui-là.

Le M10 offre une autre surprise agréable : la frappe. Le clavier, malgré la petitesse de l'appareil, est professionnel, et il est parfaitement adapté aux habitudes des pays dans lesquels il est vendu (le clavier est francisé de façon complète chez nous) avec ses 56 touches; cela donne 94 symboles spéciaux ou graphiques, avec en plus

8 touches de fonctions programmables, 4 touches de commandes et 4 touches de contrôle du curseur.

Côté mémoire, il ne faut pas s'attendre à des miracles, évidemment. D'abord, sachez que ce micro portatif s'articule autour d'un microprocesseur 8 bits 80C85 CMOS. On retrouve là une technologie propre aux calculatrices. Néanmoins, la puissance du M10 est loin d'être négligeable, avec une mémoire morte de 32 K, extensible à 64 K, et une mémoire vive de 8 ou 24 K, extensible à 32 K par modules de 8 K. Cette RAM est non volatile (les données sont conservées pendant quarante jours dans la configuration 8 K).

Le M10 est livré avec cinq logiciels intégrés : un traitement de texte (TEXT), un fichier d'adresses (ADRESS), un agenda électronique (SHEDL), un logiciel de télécommunication (TELCOM) et un BASIC Microsoft. Notons que TEXT est un logiciel relativement limité dans ses capacités. Si l'on doit souligner au passage qu'un générateur de sons est disponible (5 octaves avec contrôle en BASIC), c'est au niveau du dialogue de la machine avec toute une série de périphériques que le M10 prend toute sa valeur. Sa vocation d'ordinateur véritablement portable se situe

Le M10 est un des premiers micros véritablement portatif. Avec un poids inférieur à 2 kg, il se glisse facilement dans un attaché-case. (Cl. Olivetti.)





La lisibilité est souvent la difficulté principale que présentent les écrans à cristaux liquides. Le M10 d'Olivetti pallie cet handicap en offrant un écran inclinable muni d'un variateur de contraste.

(Cl. Gianalberto Cigolini.)

là. Et il faut reconnaître que l'on n'est pas déçu. Plusieurs interfaces permettent au M10 de se connecter à différents périphériques, qu'ils soient portables autonomes comme un lecteur-enregistreur de cassettes, un micro-plotter pour écrire ou dessiner en quatre couleurs (utilisation d'un papier de 114 mm de largeur et alimentation autonome à batteries rechargeables), un lecteur de codes à barres ou encore un coupleur acoustique à modem intégré; qu'ils soient des périphériques de bureau comme les machines électroniques Olivetti (interfacées) de la gamme Praxis ou ET, des imprimantes à aiguilles, à marguerites ou à jet d'encre, ou encore avec d'autres micros (un deuxième M10 ou un M20).

Il ne fait aucun doute qu'avec cet équipement la grande firme italienne Olivetti a marqué un point important dans la bataille qu'elle livre pour prendre la place de leader européen du marché de la micro-informatique. Son récent accord avec la société américaine ATT ne peut que renforcer son potentiel technologique et son agressivité commerciale.

Sur ce dernier point, elle montre d'ailleurs qu'elle sait y faire. Avec le M10, elle propose une cassette qui guide pas à pas le nouvel acquéreur. Il suffit de suivre le mode d'emploi :

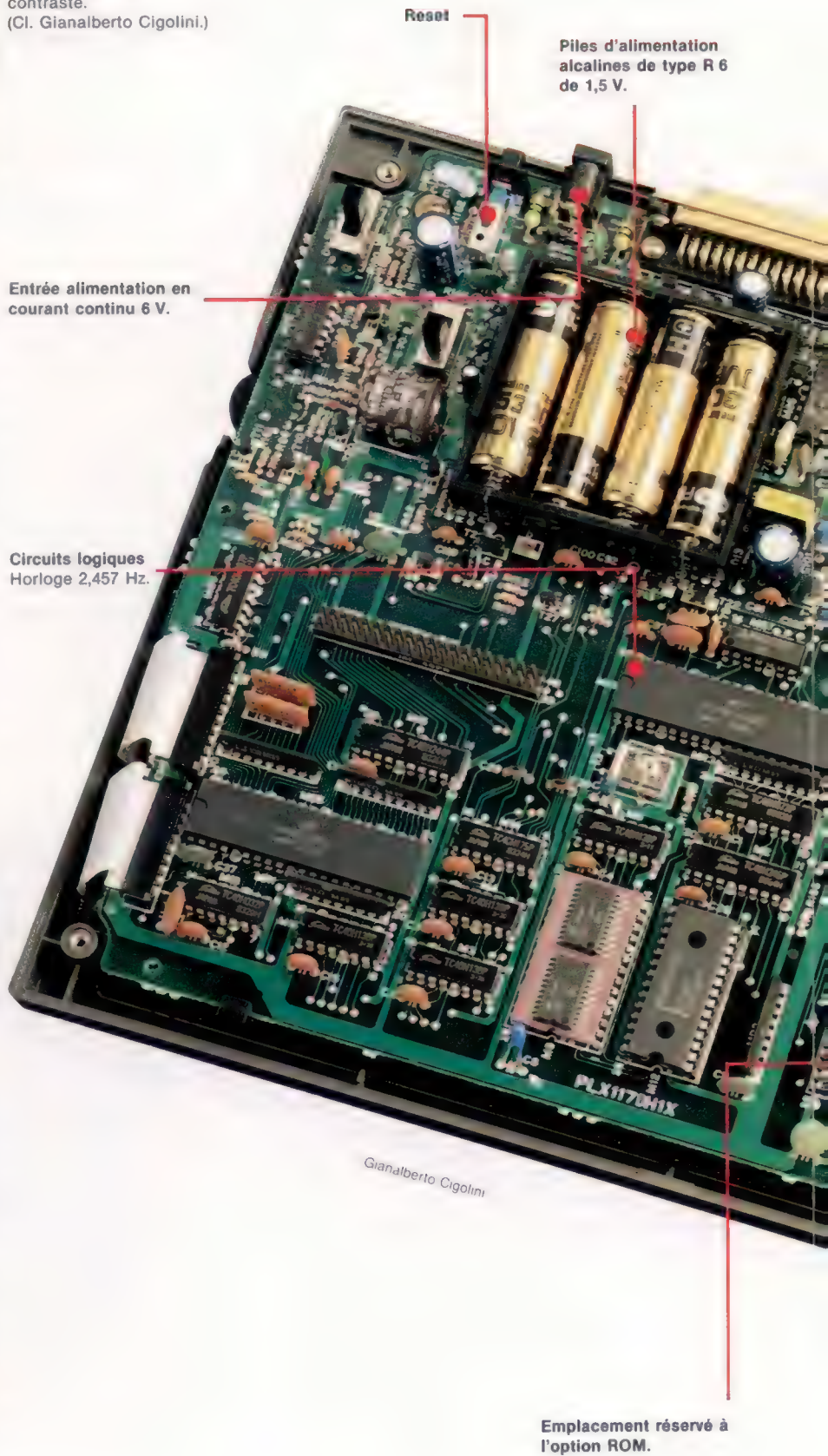
« Choisissez un magnétophone qui dispose d'une sortie écouteur (EAR), d'une prise pour microphone (MIC), d'une prise de télécommande (REMOTE), d'un bouton de réglage de tonalité, d'un autre de réglage de volume et d'un compteur (COUNTER)...

» Réglez le contrôle de la tonalité sur les aigus et le volume au 2/3 de sa puissance (vous pourrez, par la suite, affiner ces réglages indicatifs).

» Connectez le M10 au câble fourni : fiche blanche dans "EAR", rouge dans "MIC" et noire dans "REMOTE". Insérez la cassette dans votre magnétophone. Mettez le compteur du magnétophone à zéro et appuyez sur "PLAY". La bande n'avance pas. C'est normal. Maintenant, c'est le M10 qui commande.

» Allumez le M10. Le menu des programmes apparaît alors sur l'écran, le curseur placé sur le mot BASIC. Appuyez alors sur la touche "ENTER", puis ensuite, et ensuite seulement, tapez : RUN"CAS; et appuyez sur "ENTER".

Un signal acoustique de quelques secondes vous confirme le bon déroulement de la bande. C'est parti! Suivez les instructions qui défilent. »



Reset

Piles d'alimentation alcalines de type R 6 de 1,5 V.

Entrée alimentation en courant continu 6 V.

Circuits logiques  
Horloge 2,457 Hz.

Gianalberto Cigolini

Emplacement réservé à l'option ROM.



**Connecteur RS 232C.**  
Permet l'émission  
et la réception des  
données.

**Connecteur pour  
imprimante type  
Centronics.**  
Un câble spécial (CBL/2)  
est nécessaire.

**Connecteur lecteur-  
enregistreur de cassettes.**  
Le lecteur-enregistreur  
doit disposer de trois  
fiches : écouteur, micro  
et télécommande.

**Connecteur pour le lec-  
teur de codes à barre.**

Trois emplacements  
réservés aux options  
RAM.

## M10 OLIVETTI

### PRIX

\*\*\*\*

### POIDS

1,700 kg.

### DIMENSIONS

30 x 5 x 21 cm.

### UC

80C85 CMOS.

### MÉMOIRE

32 K en ROM, 8 à 24 K en  
RAM extensible à 32 K par  
modules de 8 K chacun.

### VISUALISATION

Cristaux liquides  
inclinables; variateur de  
contraste; 8 lignes de  
40 caractères.

### INTERFACES

Parallèle pour imprimante;  
série RS 232C; cassette  
(1 500 bauds); lecteur  
codes à barre; phone.

### LANGAGE DISPONIBLE

Basic.

### CLAVIER

56 touches, 8 touches  
fonctions programmables,  
4 touches commandes,  
4 touches contrôle curseur.

### DOCUMENTATION

Manuel clair en deux  
parties : un guide  
utilisateur du M10 et un  
guide du langage BASIC qui  
récapitule la structure, la  
syntaxe et les commandes  
du BASIC sur M10.

### FORCES

Véritable micro portatif,  
connectable à de  
nombreux périphériques.

### FAIBLESSES

Capacité mémoire un peu  
juste, surtout pour un  
traitement de texte.



### Partout et ailleurs

Le M10 se veut l'outil idéal de l'homme qui bouge. Son  
dispositif d'accouplement acoustique permet de le brancher  
immédiatement sur une ligne téléphonique. (Cl. Olivetti.)

# Déflagration

Dernière touche à notre « Jeu de mines » pour le BBC Micro et l'Electron avec, en particulier, le mode télétexte du BBC Micro pour l'utiliser dans l'affichage du message « fin de partie ».

Le mode graphique 7 du BBC Micro, également appelé Télétexte, présente des caractéristiques tout à fait remarquables. Sa finalité est l'affichage d'informations transmises depuis des sources extérieures, comme Télétel, faisant intervenir un autre micro et une ligne téléphonique normale. Les autres caractéristiques graphiques que permet le mode 7 consistent en un affichage sophistiqué de texte. Quelques instructions simples suffisent et nous utiliserons ce mode pour notre affichage « fin de partie ».

Les codes de commande CHR\$( ) dans des instructions PRINT permettent de contrôler le texte et les couleurs de fond, de susciter le clignotement du texte à l'écran et de produire des caractères en double hauteur. La fonction TAB peut être utilisée de la manière habituelle pour positionner le texte sur l'écran 25 x 40. On dispose de sept couleurs que l'on choisit par les codes suivants :

129	rouge
130	vert
131	jaune
132	bleu
133	magenta
134	cyan
135	blanc

#### Attention où vous mettez les pieds !

Ces cadres représentent les écrans obtenus au cours de diverses parties. On peut y voir le personnage, le tireur embusqué, une explosion, le score et l'affichage du titre.

Lorsque l'on utilise le mode 7, le texte s'affiche en blanc sur fond noir. La couleur peut être modifiée en tout point du texte, à l'intérieur d'une instruction PRINT, en insérant un code de commande. Par exemple, les trois mots de la phrase suivante seront respectivement affichés en rouge, blanc et bleu :

```
PRINT CHR$(129);« JEUX »;CHR$(135);« SANS »;  
CHR$(132);« FRONTIERES »
```

Il est important néanmoins de comprendre que la ligne affichée (PRINT) suivante le sera en blanc, la couleur par défaut restaurée. Aussi, si nous voulons continuer l'affichage dans des couleurs différentes, il faudra spécifier les codes de commande appropriés sur chaque nouvelle ligne. Même dans le cas où il s'agit de poursuivre l'affichage dans les couleurs déjà données.

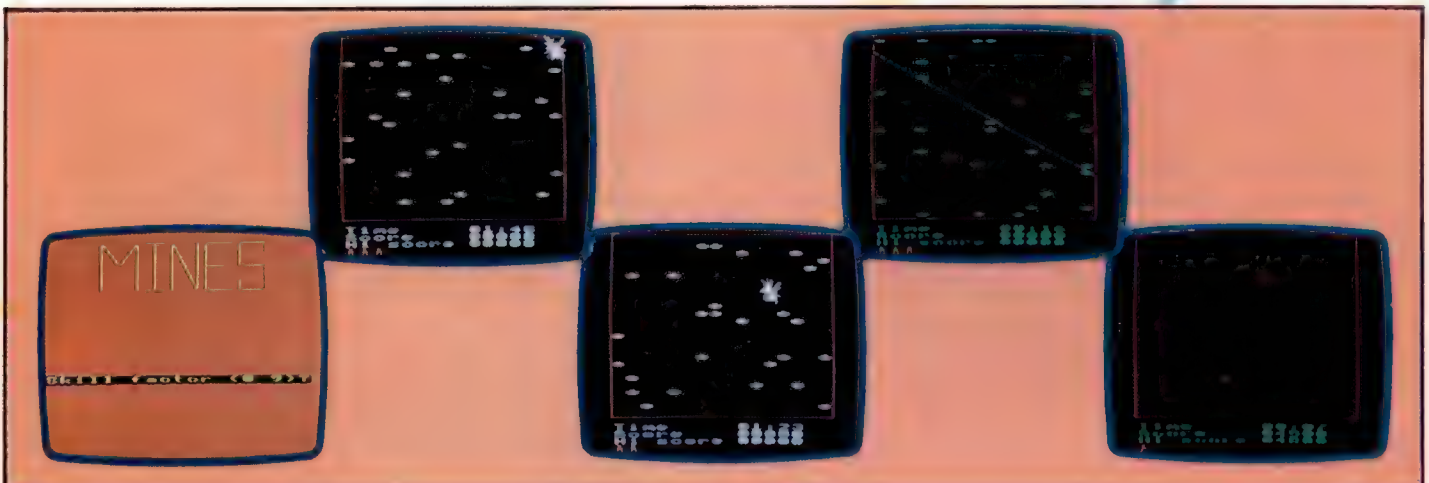
De même que nous pouvons choisir la couleur du texte, il nous est possible de déterminer la couleur du fond. Il suffit de spécifier « CHR\$(157) », suivi de la couleur voulue. Ainsi, pour produire des lettres bleues sur un fond blanc, la combinaison suivante de codes de commande sera utilisée :

```
PRINT CHR$(132);CHR$(157);CHR$(135);« LA MARINE  
FRANÇAISE... »
```

Le premier code définit la couleur du texte, les deux autres, la couleur du fond. On peut faire clignoter le texte aussi bien que le fond en plaçant un code CHR\$(136), immédiatement avant celui de la couleur. CHR\$(137) annule cet effet visuel. Nous ferons clignoter les lettres bleues ainsi :

```
PRINT CHR$(136);CHR$(132);CHR$(157);CHR$(135);« LA  
MARINE FRANÇAISE... »
```

La caractéristique la plus étonnante du mode graphique 7 du BBC Micro est de pouvoir produire des caractères en double hauteur. CHR\$(141) doit être spécifié et nous devons en outre afficher (PRINT) la même ligne deux fois de suite pour obtenir l'effet voulu. Par exemple, pour produire des caractères bleus en double hauteur, clignotant sur un fond blanc, nous utiliserons les lignes



de code suivantes :

```
10 MODE 7
20 PRINT CHR$(141);CHR$(136);CHR$(132);CHR$(157);
  CHR$(135);« LA MARINE FRANÇAISE... »
30 PRINT CHR$(141);CHR$(136);CHR$(132);CHR$(157);
  CHR$(135);« LA MARINE FRANÇAISE... »
```

Vous remarquerez que tous ces codes de commande prennent de nombreuses lignes et beaucoup de temps de frappe. Une autre méthode consiste à chaîner ensemble plusieurs codes. La chaîne obtenue peut ensuite être placée dans des instructions PRINT. Ainsi, si nous voulons afficher (PRINT) des caractères rouges sur fond jaune, nous pouvons commencer par créer une chaîne spécifique (rouges\$) qui sera par la suite utilisée dans chaque instruction PRINT appropriée :

```
10 MODE 7
20 rouges$ = CHR$(129) + CHR$(157) + CHR$(131)
30 PRINTrouges$;« LA PARTIE EST FINIE »
40 PRINTrouges$; « VOTRE SCORE »
```

Voyons en détail comment ces effets sont mis en œuvre dans la page-écran « FIN DE PARTIE » :

```
211DEF PROCfin_de_partie
212IF_score > le_plus_grand_score THEN le
  plus_grand_score=score
213rouges$=CHR$(129)+CHR$(157)+CHR$(131)
214leu$="FIN DE PARTIE"
215PRINTTAB(8;5)rouges$CHR$(141)CHR$(136)
  TAB(12)leu$
216PRINTrouges$CHR$(141)CHR$(136)ITAB(12)
  leu$
217PRINTPRINTrouges$"Votre score"ITAB(38)
  score$
218PRINTrouges$"le_plus_grand
  score"ITAB(38)le_plus_grand_score$
219PRINTPRINTrouges$TempsITAB(38)temps$
220bleu$=CHR$(132)+CHR$(157)+CHR$(134)
221autre_partie$="UNE AUTRE PARTIE O/N ?"
222PRINTPRINT
223PRINTbleu$CHR$(141)CHR$(136)ITAB(3)
  autre_partie$
224PRINTbleu$CHR$(141)CHR$(136)ITAB(3)
  autre_partie$
225REM == REPONSE ==
226= FX 15.1
227reponse$=FRAPPE(8)
228IF TROUVE="N" THEN drapeau_fin=1
229= INPROC
```

La ligne 210 vérifie si le score de la partie qui vient de s'achever est supérieur au plus grand score enregistré, et, si besoin est, l'enregistre. Le message « LA PARTIE EST FINIE » s'affiche alors en double hauteur, caractères rouges clignotants, sur fond jaune (lignes 213 à 216). Les détails des scores ainsi que les temps sont également affichés (lignes 217 à 219). Le joueur se voit alors demander s'il désire faire une autre partie. Si la réponse est Non, une variable (drapeau de fin) est mise à 1.

Remarquez que le mode 7 n'a pas été retenu pour cette procédure. La raison en est que le BBC Micro ne permet pas de changement de mode à l'intérieur d'une même procédure. Une telle tentative résulterait en un message d'erreur « MODE NON APPROPRIÉ ». Nous devons, à la place, spécifier le mode 7 dans le court programme d'appel de procédures et ajouter les lignes suivantes. Vous remarquerez que la totalité du programme d'appel a maintenant été placée dans une boucle REPEAT...UNTIL qui s'achèvera lorsque le drapeau de fin sera mis à 1.

```
1100 REPEAT
1200 MODE7
1210REM == SUPPRIMER CURSEUR ==
1220VDU23,1,0:0:0:0
1230PROCfin_de_partie
1240UNTIL drapeau_fin=1
1250CLS
1260END
```

## Une alternative

Les utilisateurs de micros Electron ont pu s'étonner dans la mesure où cet ordinateur ne comporte pas de mode 7. Pour combler cette lacune, nous avons mis au point une procédure qui utilise le mode 5 pour la page-écran de « fin de partie ». Retirez la ligne 1200 donnée plus haut et remplacez la procédure « FIN DE PARTIE » du BBC Micro par la procédure suivante :

```
211DEF PROCfin_de_partie
212IF_score > le_plus_grand_score THEN le
  plus_grand_score=score
213rem s'abaisser que le fond est bien
  jaune
214VDU19,138,3,0,0,0
215OCOL0,138COL0:REM COULEUR DE L'ECRAN
216COULEUR1,COULEUR138:REM DETERMINER LES
  COULEURS POUR LE TEXTE
217leu$="FIN DE PARTIE"
218PRINTTAB(2;4)leu$
219COULEUR0
220PRINTTAB(8;5)"VOTRE SCORE"ITAB(12)score$
221PRINTPRINTleu$plus_grand_score$
  TAB(12)leu$plus_grand_score$
222PRINTPRINTTempsITAB(12)temps$
223autre_partie$="UNE AUTRE PARTIE O/N ?"
224rem REPONSE
225VDU3,3,1,0,0,0
226COULEUR3
227PRINTPRINT
228PRINTTAB(2)autre_partie$
229REM == REPONSE ==
230= FX 15.1
231reponse$=FRAPPE(8)
232IF TROUVE="N" THEN drapeau_fin=1
233VDU28:REM RESET MANQUE COULEURS
234= ENDPROC
```

### Listage final

```
1100REPEAT
1110MODE5
1120REM == SUPPRIMER CURSEUR ==
1130VDU23,1,0:0:0:0
1140PROCfin_de_partie
1150CLS
1160PROCinitialisation
1170
1180PROCcouleur
1190
1200DEF
1210REM == SUPPRIMER CURSEUR ==
1220VDU23,1,0:0:0:0
1230PROCfin_de_partie
1240UNTIL drapeau_fin=1
1250CLS
1260END
1270
1280
1290REM == DEFINITION DES PROCEDURES ==
1300DEF PROCdebut
1310COL 8,129
1320COL0
1330COL 3,3
1340PROCcouleur
1350Y100:Y=0
1360REPEAT
1370X=X+28:Y=Y+50
1380DR 1=1 TO 2
1390PROCmes
1400HEIT 1
1410UNTIL Y 700
1420
1430PROCmes
1440PRINTTAB(8;28)"Niveau de difficulté (0-9)?"
1450PROCcouleur
1460REPEAT
1470dr=fr.couleur-TROUVE-48
1480UNTIL dr.couleur<=1-AND dr.couleur>=0
1490= ENDPROC
1500
1510DEF PROCmes
1520PLOT0,3,Y
1530REM == LETTRE M ==
1540PLOT1,0,200
1550PLOT1,0,-100
1560PLOT1,00,100
1570PLOT1,0,-200
1580PLOT0,48,0
1590PLOT1,00,0
1600PLOT0,-40,0
1610PLOT0,0,200
1620PLOT0,-40,0
1630PLOT1,00,0
1640PLOT1,00,0
1650REM == LETTRE N ==
1660PLOT0,48,-20
1670PLOT1,0,200
1680PLOT1,120,-200
1690PLOT1,0,200
1700REM == LETTRE E ==
1710PLOT0,160,0
1720PLOT1,-120,0
1730PLOT1,0,-200
```



Les variables et les procédures ont des noms très intelligibles dans le programme lui-même. Le caractère « \_ » ne doit pas être oublié pour les mots composés. Ce caractère, « \_ », apparaît par exemple dans « le\_plus\_grand\_Score\$ » ou dans « PROCfin\_de\_partie ». Il s'agit du souligné et non du trait d'union. Les utilisateurs du BBC Micro savent que c'est un séparateur légal.

```

1740PLOT1,120,0
1750PLOT0,-40,100
1760PLOT1,-80,0
1770REM ** LETTRE S
1780PLOT0,200,60
1790PLOT1,0,40
1800PLOT1,-120,0
1810PLOT1,0,-100
1820PLOT1,120,0
1830PLOT1,0,-100
1840PLOT1,-120,0
1850PLOT1,0,40
1860ENDPROC
1870:
1880DEF PROCinitialisation
1890COULEUR 2
1900drapeau_fin=0
1910PROCinitialisation
1920PROCdéfini
1930niveau_difficulté=0
1940PROCpose_de_mines(fact)
1950PROCtracer_marse
1960PROCinitialiser_temps
1970PROCinitialiser_score
1980PROCinitialiser_hommes
1990PROCposition_caractères
2000ENDPROC
2010:
2020DEF PROCboucle
2030REPEAT
2040PROCmise_à_jour_heure
2050PROCtest_cavier
2060rand=RND(500+difficulté)
2070IF rand=1 THEN PROCtir
2080 UNTIL TEMPS_12099 DR_drapeau_fin=1
2090 ENDPROC
2100:
2110DEF PROCfin_de_partie
2120 IF score$>le_plus_grand_score$ THEN le
plus_grand_score=score$
2130cuse=CHR$(129)+CHR$(157)+CHR$(131)
2140eue$="FIN DE PARTIE"
2150PRINTTAB(0,5)+cuse+CHR$(141)+CHR$(136)+TAB(12)+eue$
2160PRINTred$+CHR$(141)+CHR$(136)+TAB(12)+eue$
2170PRINT:PRINT"Voilà le score: "+TAB(30)+score$
2180PRINT:PRINT"le plus grand score: "+le_plus_grand_score$
2190PRINT:PRINT"Vous avez joué "+TAB(10)+eue$
2200eue$=CHR$(132)+CHR$(117)+CHR$(132)
2210eoc$="Une autre partie O/N"
2220PRINT:PRINT
2230PRINTeue$+CHR$(141)+CHR$(136)+TAB(5)+autre_partie$
2240PRINTeue$+CHR$(141)+CHR$(136)+TAB(5)+autre_partie$
2250REM ** REPONSE **
2260*FX 15,1
2270*raponse$=INKEY$(0)
2280IF TROUVE$="N" THEN drapeau_fin=1
2290ENDPROC
2300:
2310REM *** PROCEDURES NIVEAU 2 ***
2320DEF PROCinitialisation_variables
2330xse t=21yde t=25xhomme=17yhomme=1
2340 xdebut=120xfin=1144
2350zero$="000000"
2360ENDPROC
2370:
2380DEF PROCdéfinition_caractères
2390REM ** MINES **
2400VDU23,224,0,0,36,234,234,124,0,0
2410REM ** DETECTEUR DE MINES **
2420VDU23,225,231,195,189,36,36,189,195,231
2430REM ** PERSONNAGE **
2440VDU23,226,56,56,16,124,186,170,40,180
2450ENDPROC
2460:
2470DEF PROCtracer_marse
2480GCCL 0,1
2490MOVE 120,180
2500DRAW 120,992
2510DRAW 1152,992
2520DRAW 1152,180
2530DRAW 120,180
2540ENDPROC
2550:
2560DEF PROCpose_de_mines(nombre_de_mines)
2570REM ** CHANGER COULEUR 2 POUR VERT **
2580VDU19,2,2,0,0,0
2590FOR I=1 TO nombre_de_mines
2600PRINTTAB(RND(16)+1,RND(25)+CHR$(224)
2610NEXT I
2620ENDPROC
2630:
2640DEF PROCinitialisation_temps
2650PRINTTAB(1,27)"Temps 02:00"
2660*MS=0
2670*PROC
2680DEF PROCinitialisation_homme
2690homme=CHR$(226)+CHR$(226)+CHR$(226)
2700total=1
2710COULEUR 1
2720COULEUR 1
2730PRINTTAB(2,30)+homme$
2740COULEUR 2
2750ENDPROC
2760:
2770DEF PROCattribuer_score
2780score=0:score$="00000"
2790PRINTTAB(2,20)"Score 00000"
2800PRINTTAB(2,29)"le_plus_grand_score: "+le_plus_grand_score$
2810ENDPROC
2820:
2830DEF PROCinitialisation_caractères
2840*PRINT
2850COULEUR 2
2860ENDPROC
2870COULEUR 2
2880ENDPROC
2890:
2900DEF PROCmise_à_jour
2910sec$=STR$(((12100-TEMPS) DIV 100)MOD 60)
2920min$=STR$(((12100-TEMPS) DIV 6000)MOD 60)
2930REM ** AJOUTER ZEROS EN TETE **
2940sec$=LEFT$(zero$,2-LEN(sec$))+sec$
2950min$=LEFT$(zero$,2-LEN(min$))+min$
2960temos$min$+" "+sec$
2970PRINTTAB(11,27)+temos
2980ENDPROC
2990:
3000DEF PROCtest_cavier
3010 REM ** VERS LE HAUT ? **
3020IF INKEY(-50)=-1 THEN PROCdéplacement(0,-1)
3030REM ** VERS LE BAS ? **
3040IF INKEY(-42)=-1 THEN PROCdéplacement(0,1)
3050REM ** VERS LA DROITE ? **
3060IF INKEY(-122)=-1 THEN PROCdéplacement(1,0)
3070REM ** VERS LA GAUCHE ? **
3080IF INKEY(-26)=-1 THEN PROCdéplacement(-1,0)
3090ENDPROC
3100:
3110DEF PROC
3120*PRINT
3130*raout=ND(750)+220
3130*rin=RD(750)+220
3140*se=21*yde*(yfin-ydebut)/yfin
3150GCCL 3,3
3160PROCine
3170IF POINT(x,y)=1 THEN PROCexplosion(x,y)DEF PROCine
3180ENDPROC
3190:
3200REM *** PROCEDURES NIVEAU 3 ***
3210:
3220 DEF PROCdéplacement(delta_x,delta_y)
3230REM ** EFFACER ANCIENNES POSITIONS **
3240COULEUR 1
3250PRINTTAB(xdet,ydet);" "
3260PRINTTAB(xhomme,yhomme);" "
3270REM ** DEPLACEMENT DETECTEUR **
3280xdet=xdet+delta_x
3290ydet=ydet+delta_y
3300REM ** TEST DE LIMITES **
3310IF xdet>17 THEN xdet=17
3320IF ydet<5 THEN ydet=5
3330IF xdet<2 THEN xdet=2
3340IF ydet=1 THEN ydet=1
3350REM ** CALCUL DES COORDONNÉES HOMME **
3360xhomme=19-xdet
3370yhomme=26-ydet
3380PROCconversion(xhomme,yhomme)
3390IF POINT(xrash,yrash)=2 THEN PROCexplosion
(xrash,yrash)
3400PROCconversion(xdet,ydet)
3410IF POINT(xrash,yrash)=2 THEN PROCine trouvée
3420PROCposition_caractères
3430ENDPROC
3440:
3450DEF PROCine
3460SOUND0,-8,4,3
3470*xxdebut=ydebut
3480MOVE x,y
3490REPEAT
3500DRAW x,y
3510*xxokkyy=y+y
3520UNTIL x_kfin OR POINT(x,y)=1
3530ENDPROC
3540:
3550DEF PROCexplosion(x_explosion,y_explosion)
3560REM ** EFFET SONORE **
3570SOUND 0,-15,6,50
3580REM ** DONNER CADENCE CLIGNOTEMENT **
3590*FX9,20
3600*FX10,50
3610*FX11,100
3620*FX12,150
3630COULEUR 1
3640GCCL 0,RND(3)
3650PLOT 1,RND(100)-50,RND(100)-50
3660NEXT I
3670PROCinitialisation
3680ENDPROC
3690:
3700REM *** PROCEDURE NIVEAU 4 ***
3710:
3720 DEF PROCconversion(xcar,ycar)
3730xrash=64*xcar+32
3740yrash=1023-(32*ycar+16)
3750ENDPROC
3760:
3770DEF PROCniveau
3780REM ** SCORE **
3790COULEUR 2
3800score=150
3810score$=STR$(score)
3820score$=LEFT$(zero$,5-LEN(score$))+score$
3830ENDPROC
3840:
3850DEF PROCinitialisation
3860total=total+1
3870IF total>2 THEN drapeau_fin=1:ENDPROC
3880CLS
3890VDU19,2,2,0,0,0
390COULEUR 2
3910PROCinitialisation_variables
3920niveau_difficulté = niveau_difficulté/150
3930PROCpose_de_mines (mines_à_gauche)
3940PROCdessiner_lignes
3950PRINTTAB(2,27)"Temps"
3960PRINTTAB(2,20)"Score"
3970PRINTTAB(11,20)+score$
3980PRINTTAB(2,29)"plus_haut_score"
3990PRINTTAB(1,29)+plus_haut_score$
4000*xxdebut=ydebut
4010*xxokkyy=y+y
4020*xxdebut=ydebut
4030*xxokkyy=y+y
4040COULEUR 2
4050PROCconversion(x,y)
4060ENDPROC
4070:
4080DEF PROCmusique
4090REM ** TRAIT **
4100SON1, -8, 213, 5
4110SON1, -8, 209, 5
4120SON1, -8, 209, 5
4130SON1, -8, 213, 5
4140SON1, -8, 209, 5
4150SON1, -8, 213, 5
4160SON1, -8, 193, 5
4170SON1, -8, 205, 5
4180SON1, -8, 197, 5
4190REM ** 2EME TRAIT **
4200SON1, -8, 185, 20
4210SON1, -8, 165, 5
4220SON1, -8, 185, 5
4230SON1, -8, 193, 20
4240REM ** 3EME TRAIT **
4250SON1, -8, 165, 5
4260SON1, -8, 193, 5
4270SON1, -8, 197, 20
4280ENDPROC

```





# Cavaliers de minuit

**Lords of Midnight se présente comme un nouveau type de jeu, en raison du mélange qu'il opère entre jeu d'aventures et « wargame ». Et il offre des milliers de vues différentes.**

Les « wargames » sur ordinateur, dérivés de jeux de plateau comme Blitzkrieg ou Diplomacy, sont très différents des jeux d'aventures proprement dits, qui se rattachent tous plus ou moins au jeu de « Donjons et Dragons ». Et leurs principes sont différents : un « wargame » nécessite une réflexion stratégique et tactique, la mise en œuvre d'armées, de fournitures, d'armements, qui doivent être représentés sur une carte du champ de bataille. Un jeu d'aventures est lié à l'effet de surprise, à l'esprit de ressource : le joueur doit s'efforcer de résoudre des problèmes successifs au fur et à mesure que l'action progresse. Ici, l'éditeur a voulu créer un nouveau type de jeu en combinant ces deux formes très structurées.

Le jeu est destiné au Spectrum 48 K. La cassette est accompagnée d'une brochure de trente pages, où vous trouverez une carte du pays de Minuit, lieu de l'action. Vous jouez le rôle de Luxor, le prince de la Lune, seigneur des Hommes libres. Il possède l'anneau de Lune, un objet qui lui permet de contrôler les actions de ses quatre compagnons (et de voir à travers leurs yeux), ainsi que de tout autre seigneur qu'il pourra recruter. Les Hommes libres contrôlent la plus grande part de la partie sud du pays, qu'ils doivent pourtant défendre contre les attaques de Doomdark, le roi-sorcier du pays de Minuit. C'est l'ordinateur qui contrôle ses troupes, installées dans une citadelle du nord, et qui entendent bien envahir le sud. Doomdark est aidé par la « couronne de glace », l'instrument magique qui jette la frayeur parmi ses ennemis. Seul Morkin, votre compagnon, est immunisé contre son pouvoir dévastateur. C'est donc lui qui devra détruire cette couronne !

Les « wargames » sont, traditionnellement, joués sur une carte qui permet à chaque joueur de voir quelles sont les forces réunies contre lui.

Dans Lords of Midnight, une telle carte, constamment remise à jour, n'est pas accessible directement ; mais elle est conservée en mémoire par l'ordinateur. Ce que vous pouvez en voir, vous le verrez à travers les yeux d'un seigneur des Hommes libres, qui peut regarder dans huit directions différentes — on en choisit une à partir du clavier. C'est pourquoi, si l'armée ennemie se cache par-delà les collines, vous serez incapable de la voir tant que vous ne serez pas allé regarder de plus près. Cela ajoute une dimension supplémentaire au jeu.

Le graphisme, superbe, est la plus grande réussite de l'ensemble. Beyond Software parle de 32 000 paysages différents, qu'on peut contem-

pler par les yeux de 32 personnages différents. Personne n'a, sans doute, songé à les dénombrer... Le Spectrum ne peut évidemment conserver en mémoire autant d'écrans ; la firme a donc développé une technique qui permet la création de paysages. Ceux-ci se réduisent tous à quinze formes fondamentales, dont chacune existe en quatre tailles différentes, pour tenir compte de la perspective. Leur combinaison autorise la création de paysages très complexes — collines, forêts, montagnes, armées, villages, citadelles —, tous représentés avec beaucoup de détails. La mémoire du Spectrum abrite une carte qui précise la position de chacun des éléments, dans n'importe lequel des quatre mille lieux où se déroule l'action. L'ordinateur peut ainsi calculer la vue que peut avoir tel ou tel personnage depuis tel ou tel point.

Lords of Midnight est remarquablement conçu et présenté. La firme a même redessiné le jeu de caractères du Spectrum (en lettres dites « à l'onciale ») afin de donner au texte une véritable allure médiévale. De plus, alors que beaucoup de programmes dissimulent leur médiocrité derrière des graphismes très réussis, ce logiciel saura vous passionner, et vous y reviendrez certainement !

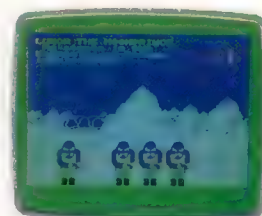
**Lords of Midnight : Spectrum 48 K.**

**Éditeur : Beyond Software.**

**Auteur : Mike Singleton.**

**Manche à balai : inutile.**

**Format : Cassette.**



Liz Heaney

## Une atmosphère médiévale

Selon Beyond Software, le jeu comporte 32 000 paysages différents — mais personne ne les a encore comptés ! Notez les lettres « à l'onciale ». Le jeu de caractères a été en effet redessiné pour donner au programme une atmosphère un peu « teutonique ».

## Une carte dans son jeu

Lords of Midnight est accompagné d'un cache qui se pose sur le clavier, et qui indique au joueur quelles sont les commandes à sa disposition.

(Cl. Liz Heaney.)

# Le tracé s'épaissit

Nous avons développé un certain nombre de routines en langage machine, qui utilisent les capacités haute résolution du Commodore 64. Nous concluons ici cette série.

Construire des algorithmes, pour remplir des formes à l'écran, peut sembler simple. En fait, les formes comprenant des angles rentrants (angles supérieurs à 180 degrés) ou les formes contractées puis dilatées présentent des difficultés particulières. Il est possible de créer une routine unique pour surmonter certaines de ces difficultés, mais pas toutes.

Donner à un programme suffisamment d'intelligence — dans la mesure où l'on peut associer ce terme à un outil informatique — pour qu'il puisse fixer de lui-même la quantité de ce qui va constituer une forme fermée n'est pas une chose facile à mettre au point, cela ne fait aucun doute.

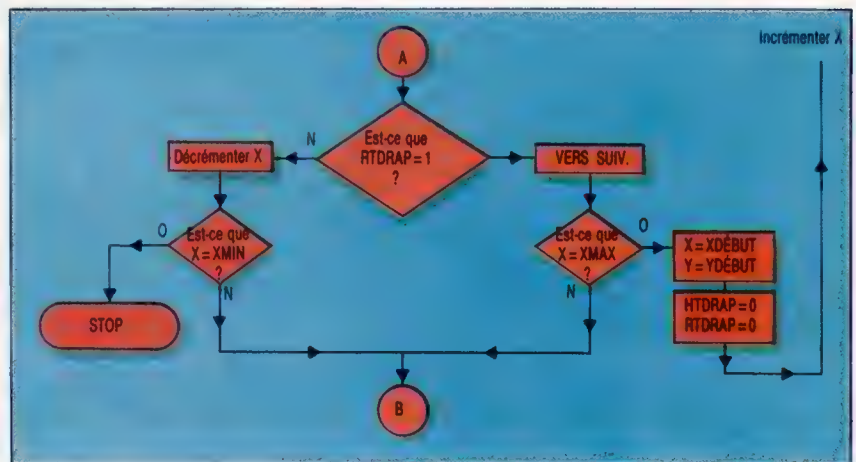
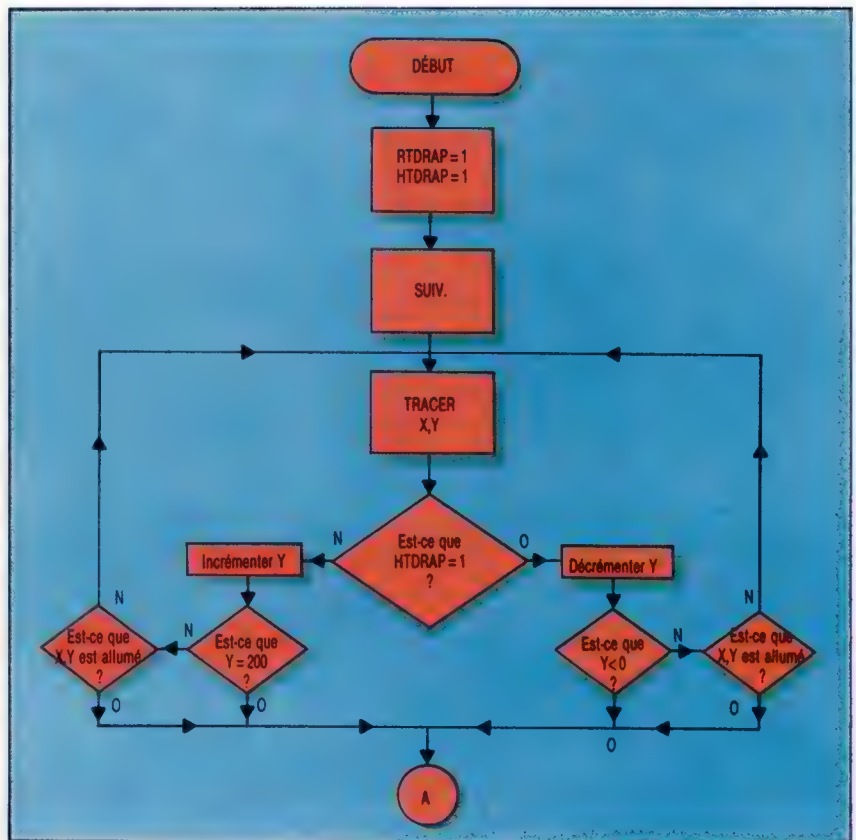
La méthode que nous allons utiliser commence par remplir la forme à partir d'un point quelconque intérieur désigné par l'utilisateur. On avance ensuite vers le haut de l'écran, jusqu'à ce qu'une frontière soit atteinte, puis on décale d'un pixel vers la droite, et l'on descend sur l'écran jusqu'à une autre frontière. Encore une fois, un décalage d'un pixel vers la droite est effectué, puis on remonte. Cette procédure se poursuit jusqu'à ce que l'extrémité droite de la forme soit remplie. Tout le processus est ensuite répété, en commençant par le point de départ et vers la gauche, jusqu'à ce que toute la forme soit remplie.

La première partie de la routine est simple. On utilise deux drapeaux, HTDRAP et RTDRAP, pour indiquer la direction de déplacement dans laquelle la route de remplissage pourra évoluer à chaque étape du programme.

La première partie de l'organigramme représente le test d'incrémentation de la routine de tracé. La boucle principale incrémente ou décrémente la valeur de la coordonnée Y selon l'état de HTDRAP. Après un test portant sur un bord de l'écran, on détermine quel est le prochain point à allumer, puis une boucle ramène au tracé du point suivant. Si l'on rencontre un pixel allumé ou un bord d'écran, la routine passe à l'étape suivante.

Le déplacement se fera ensuite soit à droite, soit à gauche, selon l'état de RTDRAP. Il est difficile de détecter la limite gauche ou droite de la forme. Au lieu de cela, la routine permet à l'utilisateur de fixer les valeurs maximale et minimale pour la coordonnée X. Cette possibilité permet, de plus, à l'utilisateur de remplir la forme avec des bandes.

Le second organigramme montre que si X est incrément (donc déplacement vers la droite) et atteint son maximum, alors les valeurs de X et Y sont réinitialisées aux valeurs de départ, et les dra-

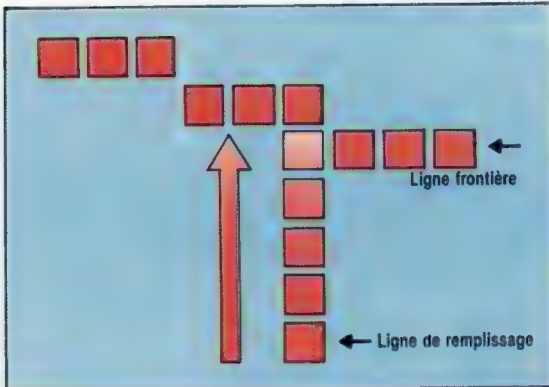


peaux de direction sont tous deux mis à zéro en vue de remplir la partie gauche de la forme. Si X est décrément et atteint XMIN, alors la routine est achevée. Si le maximum ou le minimum de X n'ont pas été atteints, alors la routine doit remplir la ligne suivante.

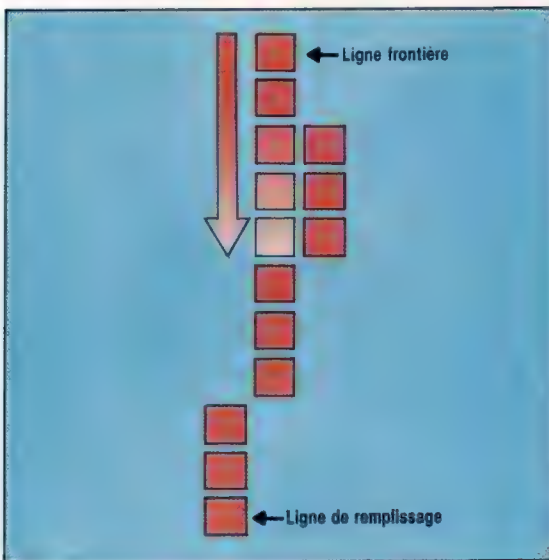


## Problèmes de gradients

Le principe de fonctionnement de l'algorithme pour traiter tous les différents types de lignes — pente faible ou forte, trait épais ou fin — est simple. Considérons la situation où la routine de remplissage se déplace vers le haut de l'écran et rencontre une ligne de faible gradient.



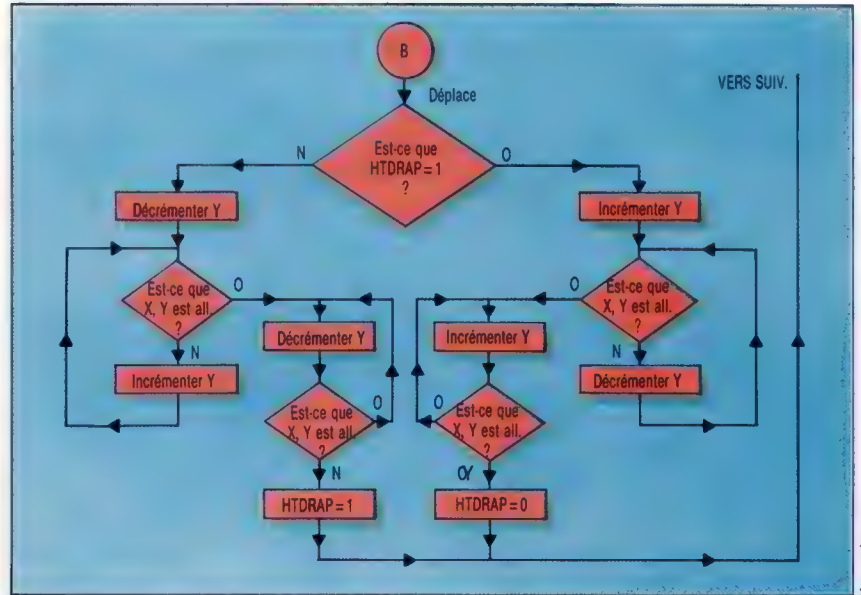
Le premier effet de l'algorithme est de retourner d'un pixel. Si le mouvement se fait vers la droite, un déplacement supplémentaire sera nécessaire afin que la ligne de remplissage suivante commence sur une position éteinte. Si, en allant vers le bas, on rencontre une pente de fort gradient, un autre problème est à surmonter.



Si le remplissage se fait vers la droite, il faut reculer de trois pixels avant de trouver un espace à remplir. Si toutefois le remplissage se fait vers la gauche, alors il faudra encore descendre de trois pixels pour atteindre la première position éteinte au-dessus de la courbe. Pour chaque direction de remplissage, il nous faut donc deux boucles dans chaque sens, avant de pouvoir localiser la première position éteinte sur la ligne de remplissage.

## Listage de « Remplisub »

La partie principale de « Remplisub » suit d'assez près l'organigramme que nous avons tracé. Les labels utilisés dans le listage code source ont été



inclus dans les positions correspondantes de l'organigramme pour vous aider à vous y retrouver dans le programme. En plusieurs endroits, il nous faut tester si un point est allumé ou non. Pour cela le programme utilise un sous-programme appelé Point, qui calcule l'adresse du point à partir de ses coordonnées X, Y de la même manière que notre programme Plotsub. Mais au lieu d'utiliser un OU logique pour mettre un bit à 1, Point emploie le ET logique pour voir si le bit en question est à 1 ou non. Si le résultat de l'opération ET n'est pas nul, alors le bit est à 1. Ce résultat est stocké en PTDRAIP pour d'autres tests dans le programme principal. Point fait en grande partie la même chose que Plotsub, et les lecteurs familiarisés avec le langage machine 6502 pourront tenter de modifier Plotsub pour qu'il puisse non seulement allumer un bit, mais aussi tester la valeur de n'importe lequel.

Enfin, voyons comment on utilise la routine Remplisub. Tout d'abord, Remplisub requiert plusieurs paramètres :

- Les coordonnées du point de départ. Ce dernier *doit* être à l'intérieur de la courbe que nous voulons remplir !
- Le maximum et le minimum de la coordonnée X. Attention aux angles très aigus.



### Mise en forme

Après un certain nombre d'articles, et après avoir tapé plusieurs kilo-octets de code, nous avons fini par réaliser sur le Commodore 64 le type de graphisme haute résolution que l'on trouve généralement comme instructions BASIC standard sur presque tous les autres micros familiaux disponibles. Si l'on veut éviter ce genre d'effort, bien sûr, il faudra dépenser 300 à 500 F pour une cartouche de BASIC étendu.



Notez que la coordonnée X du point de départ et les limites de X doivent être scindées sous la forme octet-hi/octet-lo, comme il est montré dans le programme de démonstration.

Quoique Remplisub ne s'appuie directement sur aucune des autres routines que nous avons développées pour le Commodore 64, les trois autres routines (Plotsub, Linesub et Circsub) sont chargées par le programme de démonstration de tracer des lignes à remplir par Remplisub.

La première fois que cette routine fut créée, le remplissage se faisait dans la direction horizontale, et non verticale. Mais on a trouvé qu'en utilisant des barres verticales pour remplir la forme, on diminuait considérablement le temps d'exécution.

### Chargeur Plotsub/II

Voici une version modifiée de la routine Plotsub donnée précédemment. Utilisez-la pour créer un nouveau fichier objet appelé « Plotsub.Hex » sur cassette ou disquette.

```
10 FORI=49408TO49408+314
20 READA:POKEI,A:IS=6+A:INEXT
30 READCC:IFCC<>THENPRINT"CHECKSUM ERROR"
100 DATA1,0,3,6,0,5,0,6,5,5,6,9,30,2
110 DATA72,138,72,152,72,173,0,193,240
120 DATA83,169,0,133,251,169,4,133,252
130 DATA162,3,160,0,173,2,193,145,251
140 DATA136,208,251,238,252,202,48,8
150 DATA208,244,145,251,160,231,208
160 DATA238,173,1,193,240,24,169,0,133
170 DATA251,169,32,133,252,162,32,160
180 DATA0,169,0,145,251,136,208,251
190 DATA238,252,202,208,246,173,24,208
200 DATA41,240,9,8,141,24,208,173,17
210 DATA208,9,32,141,17,208,76,125,193
220 DATA173,24,208,41,240,9,4,141,24
230 DATA208,173,17,208,41,223,141,17
240 DATA208,104,168,104,178,104,96,72
250 DATA138,72,152,72,173,4,193,141,7
260 DATA193,173,3,193,41,240,141,6,193
270 DATA173,3,193,41,7,141,8,193,173,5
280 DATA193,41,7,141,10,193,162,3,78,5
290 DATA193,202,208,250,173,5,193,141
300 DATA9,193,169,0,141,11,193,141,12
310 DATA193,162,5,173,11,193,24,109,9
320 DATA193,141,11,193,202,208,243,162
330 DATA6,14,12,193,14,11,193,144,3
340 DATA238,12,193,202,208,242,173,11
350 DATA193,24,109,6,193,141,11,193
360 DATA173,12,193,109,7,193,141,12
370 DATA193,173,11,193,24,105,0,141,11
380 DATA193,173,12,193,193,32,141,12
390 DATA193,173,11,193,24,109,10,193
400 DATA141,11,193,173,12,193,105,0
410 DATA141,12,193,173,11,193,133,251
420 DATA173,12,193,133,252,169,1,141
430 DATA13,193,56,169,7,237,8,193,240
440 DATA7,170,14,13,193,202,208,248
450 DATA160,0,177,251,13,13,193,145
460 DATA251,76,125,193
470 DATA37523:REM#CHECKSUM#
```

**Étrange dispositif**  
La ligne 15 DN=8 indique que les fichiers objets (Plotsub.Hex, etc.) doivent être chargés à partir d'une disquette. En cas d'utilisation de cassette, remplacer par DN=1, et faites une bande avec les fichiers objets dans l'ordre spécifié par les lignes 20 à 30 ou, si vos fichiers sont sur différentes cassettes, insérez le code suivant aux lignes 22, 26 et 28 :  
INPUT « CHANGER CASSETTE ET APPUYER SUR 'RETURN' », A\$

### Remplisub Demo

```
10 REM **** PROGR. REMPLISUB. DEMO ****
15 DN=0:REM POUR CASSETTE DN=1
20 IFA=0THENA=1:LOAD"PLOTSUB.HEX",DN,1
25 IFA=1THENA=2:LOAD"LINESUB.HEX",DN,1
27 IFA=2THENA=3:LOAD"CIRCSUB.HEX",DN,1
30 IFA=3THENA=4:LOAD"REPLISUB.HEX",DN,1
40 GOSUB1000:REM HIRES
50 REM **** TRACE TRIANGLE ****
60 XA=100:YA=150:XB=300:YB=160:XC=170:YC=200
80 X1=XA:Y1=YA:X2=XB:Y2=YB:GOSUB2000
90 X1=XC:Y1=YC:GOSUB2000
100 X2=XA:Y2=YA:GOSUB2000
102 REM **** TRACE CERCLE ****
103 XC=60:YC=60:R=50:GOSUB4000
```

### Remplisub Demo (suite)

```
120 REM **** REML.TRIANGLE ****
130 IS=170:YS=130:RECH.POINT.DE.DEPART
140 MIN=100:MAX=300:REM LIMITE
150 REM ****
161 REM **** REML.CERCLE ****
162 YS=60:YS=60:RECH.POINT.DE.DEPART
163 MIN=10:MAX=180
170 REM ****
200 GETTAB:IFAB="":THEN200:REM ATTEND TOUCHE
210 POKE49408:8:SYS9422:REM REINIT. EDRAW
220 PRINTCHR$(147):REM EFFACE ECRAN
225 PRINT"FIN DE ROUTINE"
230 END
1000 REM **** HIRES ****
1010 POKE49408:L:POKE49409,1
1020 POKE49410,7
1030 SYS9422
1040 RETURN
2000 REM **** LINESUB ****
2010 MH=INT(X1/256):MLD=X1-256*MH
2020 MH=INT(X2/256):MLD=X2-256*MH
2030 POKE49920,MLD:POKE49921,MH
2040 POKE49922,MLD:POKE49923,MH
2050 POKE49924,Y1:POKE49925,Y2
2060 SYS 49924
2070 RETURN
3000 REM **** FILLSUB ****
3010 SH=INT(Y1/256):SL=Y1-256*SH
3020 MH=INT(X1/256):ML=X1-256*MH
3030 MH=INT(X2/256):ML=X2-256*MH
3040 MIN=INT(MIN/256):MLN=MIN-256*MLN
3050 POKE50900,SL:POKE50901,SH
3060 POKE50902,ML:POKE50903,MLN
3070 POKE50904,MLN:POKE50905,ML
3080 SYS50907
3090 RETURN
4000 REM **** CIRCSUB ****
4010 CH=INT(Y1/256):CLD=Y1-256*CH
4020 POKE50907,CLD:POKE50908,CH
4030 POKE50909,Y1:POKE50910,R
4040 SYS 50911
4050 RETURN
5100 DATA50785:REM#CHECKSUM#
```

### Chargeur Remplisub

```
10 REM **** CHARGEUR BASIC POUR REMPLISUB ****
20 FORI=50944 TO 51075
30 READA:POKEI,A:CC=CC+INEXT
40 READA:IFCC A THEN PRINT"CHECKSUM ERROR":END
100 DATA1,0,6,8,0,3,6,5,136,39,16,60
110 DATA0,6,18,0,109,0,8,1,18,0,15
120 DATA173,11,199,141,20,199,173,12
130 DATA199,141,21,199,172,13,199,169
140 DATA1,141,18,199,141,19,199,140,5
150 DATA193,173,20,199,141,3,193,173
160 DATA21,199,141,4,193,32,131,193
170 DATA173,19,199,200,200,192,200
180 DATA240,19,76,82,199,136,192,0,144
190 DATA11,32,246,199,173,22,199,200,3
200 DATA76,46,199,173,18,199,208,31
210 DATA173,20,199,56,233,1,141,20,199
220 DATA173,21,199,233,0,141,21,199
230 DATA205,15,199,208,65,173,20,199
240 DATA205,14,199,208,57,96,173,20
250 DATA199,24,105,1,141,20,199,173,21
260 DATA199,105,0,141,21,199,205,17
270 DATA199,208,34,173,20,199,205,16
280 DATA199,208,26,173,11,199,141,20
290 DATA199,173,12,199,141,21,199,172
300 DATA13,199,169,0,141,19,199,141,10
310 DATA199,76,46,199,173,19,199,200
320 DATA20,136,32,246,199,173,22,199
330 DATA208,4,200,76,191,199,230,19
340 DATA199,136,32,246,199,173,22,199
350 DATA208,247,76,46,199,200,32,246
360 DATA199,173,22,199,208,4,136,76
370 DATA213,199,206,19,199,200,32,246
380 DATA199,173,22,199,208,247,76,46
390 DATA199,72,138,72,152,72,140,2,199
400 DATA173,20,199,141,0,199,173,21
410 DATA199,141,1,199,173,1,199,141,4
420 DATA199,173,0,199,41,248,141,3,199
430 DATA173,0,199,41,7,141,5,199,173,2
440 DATA199,41,7,141,7,199,162,3,78,2
450 DATA199,202,208,250,173,2,199,141
460 DATA6,199,169,0,141,0,199,141,9
470 DATA199,162,5,173,8,199,24,109,6
480 DATA199,141,8,199,202,208,243,162
490 DATA6,14,8,199,46,9,199,202,208
500 DATA247,173,8,199,24,109,3,199,141
510 DATA0,199,173,9,199,109,4,199,141
520 DATA9,199,173,8,199,24,105,0,141,8
530 DATA199,173,9,199,105,32,141,9,199
540 DATA173,8,199,24,109,7,199,133,251
550 DATA173,9,199,105,8,133,252,169,1
560 DATA141,18,199,56,169,7,237,5,199
570 DATA240,7,178,14,18,199,202,208
580 DATA250,160,0,177,251,45,18,199
590 DATA141,22,199,104,168,104,178,104
600 DATA50785:REM#CHECKSUM#
610 DATA50785:REM#CHECKSUM#
```



## Listage d'assemblage

```

;
;*****
;*****
;+ + REMPLISUB 6A + +
;+ +
;*****
;*****
;
;++++ VALEURS PLDTSUB +++++
;
PLTSUB = %C183
XLO = %C183
XHI = %C184
YLO = %C185
MPBLO = %00
MPBHI = %20
PTR = %FB
* = %C700
;
;++++ VALEURS REMPLISUB +++++
;
PXLO +++++1
PXHI +++++1
PYLO +++++1
PHBLO +++++1
PHBHI +++++1
PREMX +++++1
PVBYTE +++++1
PREMY +++++1
PROWLO +++++1
PR OHHI +++++1
PBPOS +++++1
;
XSTLO +++++1
XSTHI +++++1
YST +++++1
XMINLO +++++1
XMINHI +++++1
XMAXLO +++++1
XMAXHI +++++1
RTDRAP +++++1
HTDRAP +++++1
FXLO +++++1
FXHI +++++1
PTDRAP +++++1
;
;++++ TRANS PT. DEPART FX.FY +++++
;
AD 08 C7 LDA XSTLO
BD 14 C7 STA FXLO
AD 0C C7 LDA XSTHI
BD 15 C7 STA FXHI
AC 0D C7 LDY YST
;
;++++ DRAPEAUX +++++
;
A9 01 LDA ##01
BD 12 C7 STA RTDRAP
BD 13 C7 STA HTDRAP
;
;++++ POINT +++++
;
NEXT
;
8C 05 C1 STY YLO
AD 14 C7 LDA FXLO
BD 03 C1 STA XLO
AD 15 C7 LDA FXHI
BD 04 C1 STA XHI
20 03 C1 JSR PLTSUB
;
;++++ INC / DEC Y COORD +++++
;
AD 13 C7 LDA HTDRAP
D0 08 BNE DECRY
C8 INY
C8 C8 CPY ##C8 ; Y EST-IL MAX ?
F0 13 BEQ NEXLIN
4C 52 C7 JMP TESTPT
;
DECRY
88 DEY
C8 00 CPY ##00 ; Y EST-IL MIN ?
90 00 BCC NEXLIN
;
TESTPT
20 F6 C7 JSR POINT ; POINT ALLUME ?
AD 15 C7 LDA PTDRAP
D0 03 BNE NEXLIN ; SI OUI BRANCHEMENT
4C 2E C7 JMP NEXT
;
;++++ DEBUT LIGNE SUIV. +++++
;
NEXLIN
AD 12 C7 LDA RTDRAP

```

```

D0 IF BNE INCRX ; TEST DRAPEAU DROIT
AD 14 C7 LDA FXLO
38 SEC
E9 01 SBC ##01
BD 14 C7 STA FXLO ; DEC X LOBYTE
AD 15 C7 LDA FXHI
BD 15 C7 STA FXHI ; DEC X HIBYTE
;
CD 0F C7 CMP XMINHI
D0 41 BNE MOVE
AD 14 C7 LDA FXLO ; X EST-IL MIN ?
CD 0F C7 CMP XMINLO
D0 39 BNE MOVE
50 RTS ; FIN DE ROUTINE
;
INCRX
AD 14 C7 LDA FXLO
18 CLC
E9 01 ADC ##01 ; INC X LOBYTE
BD 14 C7 STA FXLO
AD 15 C7 LDA FXHI
E9 00 ADC ##00 ; INC X HIBYTE
BD 15 C7 STA FXHI
;
CD 11 C7 CMP XMAXHI
D0 22 BNE MOVE
AD 14 C7 LDA FXLO ; X EST-IL MAX ?
CD 10 C7 CMP XMAXLO
D0 1A BNE MOVE
;
AD 0B C7 LDA XSTLO
BD 14 C7 STA FXLO
AD 0C C7 LDA XSTHI ; REINIT-POINT DEPART
BD 15 C7 STA FXHI
AC 0D C7 LDY YST
;
A9 00 LDA ##00
BD 13 C7 STA HTDRAP
BD 12 C7 STA RTDRAP ; REINIT-DRAPEAUX
4C 2E C7 JMP NEXT
;
;++++ TROUVE DEBUT LIGNE SUIV. +++++
;
MOVE
AD 13 C7 LDA HTDRAP
D0 1C BNE DOWN ; TEST HTDRAP
88 DEY ; Y=Y-1
AGAIN1
20 F6 C7 JSR POINT
AD 16 C7 LDA PTDRAP
D0 04 BNE CONT1
C8 INY
4C 8F C7 JMP AGAIN1
CONT1
EE 13 C7 INC HTDRAP ; MET A 1
;
88 DEY
20 F6 C7 JSR POINT
AD 16 C7 LDA PTDRAP
D0 F7 BNE AGAIN2
4C 2E C7 JMP NEXT
;
DOWN
C8 INY ; Y=Y+1
AGAIN3
20 F6 C7 JSR POINT
AD 16 C7 LDA PTDRAP
D0 04 BNE CONT2
88 DEY
4C DB C7 JMP AGAIN3
CONT2
C3 13 C7 DEC HTDRAP ; MET A 0
;
AGAIN4
C8 INY
20 F6 C7 JSR POINT
AD 16 C7 LDA PTDRAP
D0 F7 BNE AGAIN4
4C 2E C7 JMP NEXT
;
;++++ FIN DE PROG. PRINC. +++++
;
;++++ 88-PROGR. TEST POINT +++++
;
POINT
48 PHA
8A TXA
48 PHA ; MET REG. SUR PILE
98 TYA
48 PHA
;
;++++ TEST POUR BIT A 1 +++++
;
BITON
A0 00 LDY ##00
B1 F9 LDA (PTR),Y ; CHARGE CONTENU D'ADRESSE
20 0A C7 AND PBPOS
BD 16 C7 STA PTFLAG ; STOCKE RESULTAT
;
68 PLA
A8 TAY
58 PLA ; SORT REG. DE LA PILE
AA TAX
58 PLA
;
80 FA BNE POWER
;
POWER
8E 0A C7 ASL PBPOS
CA DEX
D0 FA BNE POWER
;
RTS

```



# Contrôle de qualité

**Softsel est le plus important grossiste mondial de matériels et de logiciels. Un jeu ou un programme de gestion venant des États-Unis ont toutes les chances d'être passés entre ses mains.**

Softsel offre aux détaillants un service extrêmement précieux. D'innombrables logiciels — dont certains sont très coûteux — paraissent tous les jours; les détaillants sont donc confrontés à un problème de choix. Il leur faut évaluer par leurs propres moyens chaque nouveau produit, ce qui prend énormément de temps, ou se fier à leur première impression, ce qui peut les amener à se retrouver avec des programmes invendables.

Ce type de réflexion est loin d'être gratuit. Avec tous les logiciels qui affluent sur le marché et la variété des machines — souvent incompatibles entre elles — sur lesquelles ils doivent « tourner », il est difficile pour le commerçant de se maintenir à la « hauteur » face à une demande encore très exigeante. Softsel entend résoudre le problème en étudiant en profondeur tout logiciel que la société va devoir ajouter ensuite à sa liste de produits.

Les responsables expliquent la méthode utilisée : « Un programme est d'abord examiné par notre département technique, qui s'assure qu'il est bien d'emploi aisé — bien programmé, bien documenté, avec de bons graphismes, etc. Il passe ensuite dans notre département des ventes, qui voit s'il mérite qu'on lui fasse de la publicité. »

Softsel estime que, lors d'une récente période de six mois, dix pour cent seulement des sept cents logiciels soumis à de telles épreuves ont été intégrés au catalogue de la firme.

Ce contrôle de qualité, joint à des conditions de vente intéressantes (possibilités de retours, etc.), et l'avantage de ne travailler qu'avec un seul fournisseur, ont fait le succès de Softsel auprès des détaillants. Ceux-ci pourraient sans doute payer un peu moins cher en s'adressant directement au fabricant, mais comme Softsel achète en gros auprès de ceux-ci, elle peut consentir des rabais, et la différence est donc minime. Comme le dit un des directeurs du marketing : « Le détaillant peut toujours acheter à meilleur marché, mais, à long terme, il est plus coûteux d'acheter directement aux fabricants, parce qu'il faut alors traiter avec des centaines de gens, et non avec une seule compagnie. »

## Des origines américaines

Softsel a été créée en 1980 par Robert Leff et David Wagman, qui travaillaient tous deux au traitement de données à Transaction Technology, une filiale de l'énorme compagnie financière Citicorp. Leur idée de base était bonne : leur compagnie a connu une expansion foudroyante. Quatre ans après sa création, elle emploie 350 personnes dans le monde entier, et son chiffre d'affaires a atteint 87 millions de dollars l'année dernière! Rien qu'aux États-Unis, la firme dispose de quatre entrepôts géants — à Atlanta, Chicago, Los Angeles et New York — qui proposent près de 4 500 programmes différents aux détaillants de tout le pays.

Softsel s'est installée en Grande-Bretagne en septembre 1982, très exactement à Feltham, non loin de l'aéroport d'Heathrow, et fournit 2 500 logiciels différents à des clients répartis dans toute l'Europe, et même au Moyen-Orient. Elle entend également accroître son activité en direction du monde des affaires. Les programmes de gestion représentent déjà près de la moitié du catalogue, et cette part ira sans aucun doute en croissant. Mais il ne s'agit pas, pour autant, que les jeux soient négligés.

L'avenir paraît prometteur. Softsel compte s'implanter encore davantage sur le marché européen; l'entreprise a déjà ouvert une filiale en Allemagne, et le siège principal sur le vieux continent est basé à Zurich. Il est prévu qu'avant la fin de cette année de nouvelles filiales, en France et en Italie, soient installées.



Herb Blumstein, directeur général



Simon Rhodes, directeur de marketing

**Page manquante  
(publicité)**

**Page manquante  
(publicité)**